Mixed GWR Simulation Write-up

Aaron Swoboda

November 22, 2013

What are our research questions? Basically:

1. Can we find the "true" model among the eight different possibilities with three model parameters?

2. Are there differences in the results based on the metric used?

3. What happens as we change the sample size and amount of error in the model?

4. How much does it really matter if we are concerned with coefficient estimates?

   - How well does our selected model perform as measured by beta RMSE?
   - Does our model perform better when we select the correct model?
   - Can we control for whether we selected a model with the correct spatial variation for a given parameter?

5. What about using bandwidth size as a dependent variable?

6. What happens when we use other decision tools to help with model selection? (Monte Carlo simulations and test statistics)

1 Methodology

Imagine a simple linear model,

$$Y = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2 + \epsilon. \tag{1}$$

In addition to the three variables listed above (Y, X_1, and X_2), assume we know the two dimensional geographical location for each of our N observations. Thus, our data consists of an $Nx5$ matrix, where Y may be house prices, X_1 and X_2 could be the living space and lot size associated with each house, and the final two columns determine the location of the observations (for instance, latitude and longitude, or distances north and east from a prescribed point serving as the origin).

The above simple model exemplifies spatial stationarity in the parameters. The β coefficients are constant over space. Instead, the coefficients could exhibit spatial non-stationarity, in which case one, two, or all three of the β coefficients are a function of location. This has a natural interpretation in the current real estate

example. Location matters. Location can matter in different ways. For instance, if the value of land varies over space, then we would expect the coefficient on lot size to vary over space, while it is also possible that the intercept varies over space to reflect variation in prices of similar houses in different locations.

While it is possible to parameterize the variation in coefficients, for instance researchers often (CITATION?) include a variable measuring the distance from an observation to an important amenity such as the Central Business District and then this distance variable could be interacted with variables whose value are predicted to vary over space. However, it is not implausible to believe that the variation in coefficients might not be easily parameterized (for instance, if land values are a non-monotonic function of distance). Researchers may instead interact variables with fixed effects for cities or census tracts. However, such strategies require the analyst to make assumptions that severely limit the type and degree of variation in the parameters. For instance, interaction terms with geographic boundaries assume discrete differences in the value of parameters across the boundaries, while instead the parameters may instead be a continuous function of location.

1.1   Local Regression to the Rescue?

Locally Weighted Regression (also known as Geographically Weighted Regression) is one possible solution to the challenge presented by spatially non-stationary regression coeffiecients. Locally Weighted Regression (LWR) techniques (also known as Geographically Weighted Regression) are described in detail by Cleveland and Devlin (1988), Brunsdon et al. (1998), Fotheringham et al. (2002), and others. It is a weighted least squares methodology in which regression coefficients are estimated over space as a function of the local data as described in Equation (2),

$$\hat{\beta}_i = (X'W_iX)^{-1}X'W_iY, \tag{2}$$

where X is a $N \times 2$ matrix of independent variables, W_i is the $N \times N$ weights matrix, and Y is the $N \times 1$ vector of dependent variable values. The weights matrix, W_i is a diagonal matrix where element w_{jj} denotes the weight that the jth data point will receive in the regression coefficients estimated at location i in the dataset. We employ a bi-square weights function and a k-nearest neighbor bandwidth approach as described in equation (3),

$$w_{jj} = \left[1 - \left(\frac{d_{ij}}{d_k}\right)^2\right]^2 \text{ if } d_{ij} < d_{ik}, \text{otherwise} = 0, \tag{3}$$

where d_{ij} denotes the distance between observations i and j, and d_{ik} is the distance from observation i to the kth nearest observation. This function assigns weights close to 1 for data points near observation i, weights positive but closer to zero for observations farther away, and zero for all $n - k$ observations farther away than the kth nearest observation.

A key decision in estimating LWR models is choosing the number of observations to include in the bandwidth. Often, researchers choose a bandwidth my minimizing a cross validation metric.

Perhaps the most common cross validation metric used in the literature (how many citations?) is the Leave One Out Cross Validation score (LOOCV), which is calculated as follows,

$$LOOCV = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(y - \hat{y}_{\neq i})^2}, \tag{4}$$

where $\hat{y}_{\neq i}$ represents the dependent variable estimate for observation i while excluding observation i from the regression. This prevents the observation from having undue influence in the regression with small bandwidths and overfitting the model. Such a model, while intuitively appealing, can be computationally expensive, as regressions must be estimated first while excluding individual observations to calculate the LOOCV and then again while including the observation to obtain the regression coefficients.

An alternative cross validation metric is known as the Generalized Cross Validation (GCV) score, which only requires calculating the regressions once per location and explicitly calculates the leverage each observation has over the regression coefficients. The GCV score calculation is detailed in equation (5),

$$n * \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{(n - v_1)^2}, \tag{5}$$

where $\hat{y}_i$ is the predicted dependent variable value for observation i, and v_1 can be interpreted as the "effective number of model parameters," and calculated as $v_1 = \text{tr}(\mathbf{S})$, where the matrix $\mathbf{S}$ is the "hat matrix" which maps y onto $\hat{y}$,

$$\hat{y} = \mathbf{S}y, \tag{6}$$

and each row of $\mathbf{S}$, r_i is given by:

$$r_i = X_i(X'W_iX)^{-1}X'W_i. \tag{7}$$

The GCV score is a convenient model selection metric that rewards models that provide a good fit to the data, while penalizing models with a greater number of model parameters (Loader, 1999; McMillen and Redfearn, 2010). (Paez et al., 2011; McMillen and Redfearn, 2010; McMillen, 2012).

1.2   Experimental Design

We generate data in the following format:

$$Y = \beta_0(location) + \beta_1(location) * X_1 + \beta_2(location) * X_2 + \epsilon, \tag{8}$$

where sometimes the coefficient is in fact stationary, $\beta_i(location) = \beta$, and other times it is non-stationary. With three coefficients each having the possibility of being stationary or not, there are eight different possible combinations, ranging from (stationary, stationary, stationary) to (non-stationary, non-stationary, non-stationary).

We generate data using all eight different combinations and then estimate all eight possible LWR models (assuming non-stationarity or not for each variable). We then calculate different Cross-Validation metrics and compare their values across models and bandwidths.

We have three different values for each coefficient in our DGP, no variation, some variation, and more variation.

We also change the sample size of our data as well as the variance of the model error term.

2 Simulation Results

We have seven different ways to pick the "best" model (the AIC, GCV, SCV, LOOCV, and RMSEs for the three different coefficients). Here are tables showing the relative frequency (in percentage) that each model number was selected by optimizing a given metric. Note that the columns in the following tables may not sum exactly to 100 due to rounding.

```
for (i in 1:8) {
    temp2 = which(mcOutput[, "True Model", ] == i, arr.ind = TRUE)
    temp3 = mcOutput[8:14, "Model Number", unique(temp2[, 2])]
    temp4 = factor(temp3)

    newdata = data.frame(ModelNum = temp4, Metric = factor(rownames(temp3),
        levels = rownames(temp3)))
    cat(paste("\n true model =", i, "\n"))
    cat("spatial variation...\n")
    print(models[i, ])
    print(round(table(newdata) * 100 * 7/sum(table(newdata)), 0))
}

##
##  true model = 1
## spatial variation...
##    beta0 beta1 beta2
## 1    no    no    no
##         Metric
## ModelNum AIC GCV SCV LOOCV B0RMSE B1RMSE B2RMSE
##        1   0   0   0    75      6      7      6
##        2  26  26   0     6      5     22     20
##        3  34  33   0     6     21      4     22
##        4   1   1   8     2      5      3     37
##        5  38  38   0     7     20     21      4
##        6   1   1   8     2      5     35      3
##        7   1   1   5     1     32      4      5
##        8   0   0  78     0      5      3      2
##
```

```
##  true model = 2
## spatial variation...
##    beta0 beta1 beta2
## 2   yes    no    no
##         Metric
## ModelNum AIC GCV SCV LOOCV B0RMSE B1RMSE B2RMSE
##        1   0   0   0     4      0      4      4
##        2  91  91  37    89     94     29     29
##        3   4   4   0     2      1      6     25
##        4   0   1  16     1      1      1     32
##        5   4   4   0     3      1     24      6
##        6   0   1  16     1      1     33      1
##        7   0   0   3     0      1      2      2
##        8   0   0  28     0      0      1      1
##
##  true model = 3
## spatial variation...
##    beta0 beta1 beta2
## 3    no   yes    no
##         Metric
## ModelNum AIC GCV SCV LOOCV B0RMSE B1RMSE B2RMSE
##        1   0   0   0    12     13      0     13
##        2  11  11   1     7      1      2     24
##        3  77  76   0    65     23     66     22
##        4   2   2  12     4      4      6     32
##        5   8   8   0     5     24      2      1
##        6   0   0   9     1      2      3      2
##        7   2   2   6     5     29     15      3
##        8   0   0  72     1      4      6      2
##
##  true model = 4
## spatial variation...
##    beta0 beta1 beta2
## 4   yes   yes    no
##         Metric
## ModelNum AIC GCV SCV LOOCV B0RMSE B1RMSE B2RMSE
##        1   0   0   0     2      1      0      6
##        2  70  68  29    65     65      6     29
##        3  10   9   0     8      2     34     25
##        4  16  19  25    21     25     14     34
##        5   2   2   0     2      2      4      2
##        6   0   1  13     1      2      8      1
##        7   0   0   3     1      2     22      2
##        8   0   0  30     0      2     12      1
##
##  true model = 5
```

```
## spatial variation...
##   beta0 beta1 beta2
## 5    no    no   yes
##         Metric
## ModelNum AIC GCV SCV LOOCV B0RMSE B1RMSE B2RMSE
##        1   0   0   0    12     13     13      0
##        2  11  10   1     7      1     24      2
##        3   8   8   0     4     24      1      2
##        4   0   1   9     1      1      2      3
##        5  78  77   0    66     23     23     66
##        6   1   2  12     5      5     32      6
##        7   1   2   6     4     30      3     15
##        8   0   0  73     1      4      2      6
##
##  true model = 6
## spatial variation...
##   beta0 beta1 beta2
## 6   yes    no   yes
##         Metric
## ModelNum AIC GCV SCV LOOCV B0RMSE B1RMSE B2RMSE
##        1   0   0   0     2      1      6      0
##        2  71  68  29    66     65     30      7
##        3   3   2   0     2      2      2      4
##        4   0   0  13     1      2      1      7
##        5  10   9   0     8      2     24     34
##        6  16  19  24    20     25     33     14
##        7   0   1   3     1      2      2     21
##        8   0   0  30     0      2      1     13
##
##  true model = 7
## spatial variation...
##   beta0 beta1 beta2
## 7    no   yes   yes
##         Metric
## ModelNum AIC GCV SCV LOOCV B0RMSE B1RMSE B2RMSE
##        1   0   0   0     5     13      1      1
##        2   8   8   1     6      0      2      2
##        3  25  24   0    18     25     16      2
##        4   1   2   9     3      1     12      4
##        5  25  24   0    18     25      2     17
##        6   2   2   9     3      1      5     12
##        7  39  40   8    44     31     50     51
##        8   1   1  72     3      4     11     11
##
##  true model = 8
## spatial variation...
```

```
##   beta0 beta1 beta2
## 8   yes   yes   yes
##        Metric
## ModelNum AIC GCV SCV LOOCV B0RMSE B1RMSE B2RMSE
##        1   0   0   0     1      1      1      1
##        2  60  57  23    53     48      7      7
##        3   6   5   0     5      3     15      5
##        4   9  11  19    12     14     16      8
##        5   6   5   0     5      3      5     16
##        6   9  11  19    11     14      8     15
##        7   3   3   3     4      4     32     33
##        8   6   7  35     8     14     16     15
```

The results of the previous tables must be taken with a grain of salt, as there are frequently times when a "true" model may include variation in a coefficient, but the degree of non-stationarity in the coefficient may be small. In such cases, choosing an incorrect model (such as one that keeps such a coefficient constant) may not be such a big problem.

Some patterns clearly emerge.

- The AIC and GCV metrics *never* select Model 1, even when it is the actual model.

- There are several occasions where the model/bandwidth combination with the smallest RMSE is not the "correct" model.

2.1   Coefficient Formulation

Even if an incorrect model is chosen, the model may yield accurate estimates of the coefficients. In each run of the simulation we estimated 50 different model/bandwidth combinations (seven different bandwidths for each of the seven models with at least one coefficient varying over space, plus the standard OLS model). We calculate the Root Mean Squared Error for each model and can rank these values. For instance, it is possible, and often the case, that the "wrong" model yields the most accurate estimates of a coefficient among all of the models implemented.

```r
colMat = matrix("black", 8, 3)
colMat[as.matrix(models) == "yes"] = "red"
par(oma = c(0, 2, 3, 0))
par(mar = rep(1, 4))
for (i in 1:8) {
    # i = 1
    layout(matrix(1:12, 4, 3, byrow = T))
    temp2 = which(mcOutput[1, "True Model", ] == i)
    inputMetrics = c("AIC", "GCV", "SCV", "LOOCV")
    inputStats = c("B0RMSE Rank", "B1RMSE Rank", "B2RMSE Rank")
    for (j in 1:4) {
```

```r
        for (k in 1:3) {
            temp3 = mcOutput[inputMetrics[j], inputStats[k], temp2]
            # summary(temp3)
            hist(temp3, xlim = c(0, 50), breaks = 5 * (0:10), col = ifelse(colMat[i,
                k] == "red", "red", "grey85"), axes = F, ylab = "rel freq",
                xlab = "rank", main = "")
        }
    }
    mtext(text = paste0("Model #", i, "(non-stationary parameters in red"),
        side = 3, outer = TRUE, line = 1.8)
    mtext(inputMetrics, at = seq(0.875, 0.125, length.out = 4), side = 2, outer = T)
    mtext(paste0("B", 0:2), at = seq(0.17, 0.83, length.out = 3), side = 3,
        outer = T, line = 0, col = colMat[i, ])
}
```

3   Accuracy of Beta Estimates

During the experiment we calculated the Root Mean Squared Error the estimated coefficients for all models. We then ranked each model from smallest to largest RMSEs and collected the RMSE values and ranks for all models that minimized one of the model selection criteria and the model with the smallest RMSE for this coefficient.

Let's see if there is a relationship between the RMSE values/ranks and the parameters we modified in the experiment.

```
# temp2 = which(mcOutput[1, 'True Model', ] == i) inputMetrics = c('AIC',
# 'GCV', 'SCV', 'LOOCV') inputStats = c('B0RMSE Rank', 'B1RMSE Rank',
# 'B2RMSE Rank') temp3 = mcOutput[inputMetrics[j], inputStats[k], temp2]
myVars = c("B0RMSE", "B1RMSE", "B2RMSE", "B0RMSE Rank", "B1RMSE Rank", "B2RMSE Rank",
    "Sample Size", "Error", "True Model", "B0 SpVar", "B1 SpVar", "B2 SpVar",
    "Model Number")
rankData1 = as.data.frame(t(mcOutput["AIC", myVars, ]))
rankData1$Metric = "AIC"

rankData2 = as.data.frame(t(mcOutput["GCV", myVars, ]))
rankData2$Metric = "GCV"

rankData3 = as.data.frame(t(mcOutput["SCV", myVars, ]))
rankData3$Metric = "SCV"

rankData4 = as.data.frame(t(mcOutput["LOOCV", myVars, ]))
rankData4$Metric = "LOOCV"

rankData = rbind(rankData1, rankData2, rankData3, rankData4)

rankData$samplesize = rankData[, "Sample Size"]
rankData$error = as.factor(rankData$Error)
rankData$truemodel = as.factor(rankData[, "True Model"])
rankData$B0sv = as.factor(rankData[, "B0 SpVar"])
rankData$B1sv = as.factor(rankData[, "B1 SpVar"])
rankData$B2sv = as.factor(rankData[, "B2 SpVar"])
rankData$metric = as.factor(rankData$Metric)
names(rankData)[which(names(rankData) == "Model Number")] = "selectedmodel"
```

3.1 RMSE Values

```
RMSEregression0 <- lm(B0RMSE ~ samplesize + metric + B0sv + B1sv + B2sv + error,
    data = rankData)
summary(RMSEregression0)

##
## Call:
## lm(formula = B0RMSE ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error, data = rankData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -0.7206 -0.1046 -0.0168  0.0750  2.6470
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.24e-01   1.32e-03   93.81   <2e-16 ***
## samplesize  -3.01e-04   1.04e-06 -290.47   <2e-16 ***
## metricGCV    3.70e-03   9.95e-04    3.72    2e-04 ***
## metricLOOCV  1.04e-02   9.95e-04   10.46   <2e-16 ***
## metricSCV    2.01e-01   9.95e-04  201.63   <2e-16 ***
## B0sv1        2.97e-01   8.61e-04  345.32   <2e-16 ***
## B0sv3        6.45e-01   8.61e-04  748.50   <2e-16 ***
## B1sv1        1.34e-02   8.61e-04   15.56   <2e-16 ***
## B1sv3        6.31e-02   8.61e-04   73.30   <2e-16 ***
## B2sv1        1.36e-02   8.61e-04   15.79   <2e-16 ***
## B2sv3        6.48e-02   8.61e-04   75.24   <2e-16 ***
## error1       1.14e-01   8.61e-04  132.67   <2e-16 ***
## error2       3.12e-01   8.61e-04  362.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.179 on 259187 degrees of freedom
## Multiple R-squared:  0.766,Adjusted R-squared:  0.766
## F-statistic: 7.09e+04 on 12 and 259187 DF,  p-value: <2e-16
```

```
RMSEregression1 <- lm(B1RMSE ~ samplesize + metric + B0sv + B1sv + B2sv + error,
    data = rankData)
summary(RMSEregression1)
```

```
##
## Call:
## lm(formula = B1RMSE ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error, data = rankData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.866 -0.166 -0.025  0.122  5.309
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.73e-01   2.12e-03   81.83  < 2e-16 ***
## samplesize  -4.02e-04   1.66e-06 -242.18  < 2e-16 ***
## metricGCV    1.26e-02   1.59e-03    7.89  3.0e-15 ***
## metricLOOCV  2.13e-02   1.59e-03   13.38  < 2e-16 ***
## metricSCV    1.80e-01   1.59e-03  113.11  < 2e-16 ***
## B0sv1        1.16e-01   1.38e-03   84.16  < 2e-16 ***
```

```
## B0sv3         1.84e-01   1.38e-03   133.24   < 2e-16 ***
## B1sv1         1.43e-01   1.38e-03   103.45   < 2e-16 ***
## B1sv3         4.13e-01   1.38e-03   299.84   < 2e-16 ***
## B2sv1         6.25e-03   1.38e-03     4.54   5.6e-06 ***
## B2sv3         1.32e-02   1.38e-03     9.57   < 2e-16 ***
## error1        1.32e-01   1.38e-03    95.95   < 2e-16 ***
## error2        3.69e-01   1.38e-03   267.90   < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.286 on 259187 degrees of freedom
## Multiple R-squared:  0.501,Adjusted R-squared:  0.501
## F-statistic: 2.17e+04 on 12 and 259187 DF,  p-value: <2e-16
```

```r
RMSEregression2 <- lm(B2RMSE ~ samplesize + metric + B0sv + B1sv + B2sv + error,
    data = rankData)
summary(RMSEregression2)
```

```
##
## Call:
## lm(formula = B2RMSE ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error, data = rankData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.901 -0.167 -0.025  0.123  4.371
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.73e-01   2.13e-03    81.35   < 2e-16 ***
## samplesize  -4.01e-04   1.67e-06  -240.17   < 2e-16 ***
## metricGCV    1.29e-02   1.60e-03     8.08   6.5e-16 ***
## metricLOOCV  2.21e-02   1.60e-03    13.82   < 2e-16 ***
## metricSCV    1.80e-01   1.60e-03   112.14   < 2e-16 ***
## B0sv1        1.13e-01   1.39e-03    81.25   < 2e-16 ***
## B0sv3        1.82e-01   1.39e-03   131.29   < 2e-16 ***
## B1sv1        8.10e-03   1.39e-03     5.84   5.1e-09 ***
## B1sv3        1.29e-02   1.39e-03     9.32   < 2e-16 ***
## B2sv1        1.43e-01   1.39e-03   103.24   < 2e-16 ***
## B2sv3        4.14e-01   1.39e-03   298.96   < 2e-16 ***
## error1       1.32e-01   1.39e-03    95.41   < 2e-16 ***
## error2       3.67e-01   1.39e-03   265.00   < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.288 on 259187 degrees of freedom
## Multiple R-squared:  0.497,Adjusted R-squared:  0.497
## F-statistic: 2.14e+04 on 12 and 259187 DF,  p-value: <2e-16
```

Now I'd like to add a variable along the lines of "does the selected model correctly identify whether the variable is non-stationary?"

```
TrueModelBeta0sv = c(2, 4, 6, 8)
TrueModelBeta1sv = c(3, 4, 7, 8)
TrueModelBeta2sv = 5:8

ModelBetaSV.selected.true = function(TrueModelBetasv) {
    temp = rankData$truemodel %in% TrueModelBetasv
    temp2 = rankData$selectedmodel %in% TrueModelBetasv
    ModelBetaSVCorrect <- temp == temp2
    ModelBetaSVCorrect
}

rankData$ModelBeta0svTrue = ModelBetaSV.selected.true(TrueModelBeta0sv)
rankData$ModelBeta1svTrue = ModelBetaSV.selected.true(TrueModelBeta1sv)
rankData$ModelBeta2svTrue = ModelBetaSV.selected.true(TrueModelBeta2sv)
```

```
RMSEregression0 <- lm(B0RMSE ~ samplesize + metric + B0sv + B1sv + B2sv + error +
    ModelBeta0svTrue, data = rankData)
summary(RMSEregression0)

##
## Call:
## lm(formula = B0RMSE ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error + ModelBeta0svTrue, data = rankData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6612 -0.1027 -0.0178  0.0727  2.6565
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.91e-01   1.48e-03  128.91  < 2e-16 ***
## samplesize       -2.87e-04   1.03e-06 -278.73  < 2e-16 ***
## metricGCV         4.23e-03   9.78e-04    4.33  1.5e-05 ***
## metricLOOCV       1.04e-02   9.78e-04   10.64  < 2e-16 ***
## metricSCV         1.82e-01   9.97e-04  182.76  < 2e-16 ***
## B0sv1             3.10e-01   8.58e-04  361.75  < 2e-16 ***
## B0sv3             6.75e-01   9.05e-04  745.99  < 2e-16 ***
## B1sv1             1.33e-02   8.47e-04   15.66  < 2e-16 ***
```

```
## B1sv3                6.20e-02   8.47e-04    73.18  < 2e-16 ***
## B2sv1                1.35e-02   8.47e-04    15.89  < 2e-16 ***
## B2sv3                6.36e-02   8.47e-04    75.08  < 2e-16 ***
## error1               1.09e-01   8.49e-04   128.32  < 2e-16 ***
## error2               2.99e-01   8.58e-04   348.72  < 2e-16 ***
## ModelBeta0svTrueTRUE -9.40e-02  9.96e-04   -94.43  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.176 on 259186 degrees of freedom
## Multiple R-squared:  0.774,Adjusted R-squared:  0.774
## F-statistic: 6.84e+04 on 13 and 259186 DF,  p-value: <2e-16
```

```
RMSEregression1 <- lm(B1RMSE ~ samplesize + metric + B0sv + B1sv + B2sv + error +
    ModelBeta1svTrue, data = rankData)
summary(RMSEregression1)

##
## Call:
## lm(formula = B1RMSE ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error + ModelBeta1svTrue, data = rankData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.820 -0.161 -0.026  0.113  5.368
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.60e-01   2.40e-03  108.37  < 2e-16 ***
## samplesize          -3.86e-04   1.66e-06 -232.97  < 2e-16 ***
## metricGCV            1.36e-02   1.57e-03    8.66  < 2e-16 ***
## metricLOOCV          2.37e-02   1.57e-03   15.02  < 2e-16 ***
## metricSCV            1.81e-01   1.57e-03  115.02  < 2e-16 ***
## B0sv1                1.04e-01   1.37e-03   75.39  < 2e-16 ***
## B0sv3                1.57e-01   1.41e-03  111.16  < 2e-16 ***
## B1sv1                9.77e-02   1.49e-03   65.51  < 2e-16 ***
## B1sv3                3.99e-01   1.38e-03  290.14  < 2e-16 ***
## B2sv1                5.86e-03   1.36e-03    4.30  1.7e-05 ***
## B2sv3                1.12e-02   1.36e-03    8.24  < 2e-16 ***
## error1               1.26e-01   1.37e-03   91.92  < 2e-16 ***
## error2               3.57e-01   1.37e-03  259.78  < 2e-16 ***
## ModelBeta1svTrueTRUE -9.59e-02  1.29e-03  -74.17  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.283 on 259186 degrees of freedom
## Multiple R-squared:  0.512,Adjusted R-squared:  0.512
## F-statistic: 2.09e+04 on 13 and 259186 DF,  p-value: <2e-16
```

```
RMSEregression2 <- lm(B2RMSE ~ samplesize + metric + B0sv + B1sv + B2sv + error +
    ModelBeta2svTrue, data = rankData)
summary(RMSEregression2)

##
## Call:
## lm(formula = B2RMSE ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error + ModelBeta2svTrue, data = rankData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -0.857 -0.162 -0.026  0.113  4.462
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.60e-01   2.41e-03  107.81  < 2e-16 ***
## samplesize         -3.85e-04   1.67e-06 -231.09  < 2e-16 ***
## metricGCV           1.40e-02   1.58e-03    8.84  < 2e-16 ***
## metricLOOCV         2.45e-02   1.58e-03   15.47  < 2e-16 ***
## metricSCV           1.81e-01   1.58e-03  114.11  < 2e-16 ***
## B0sv1               1.00e-01   1.38e-03   72.30  < 2e-16 ***
## B0sv3               1.55e-01   1.42e-03  109.58  < 2e-16 ***
## B1sv1               8.18e-03   1.37e-03    5.96  2.5e-09 ***
## B1sv3               1.15e-02   1.37e-03    8.38  < 2e-16 ***
## B2sv1               9.82e-02   1.50e-03   65.45  < 2e-16 ***
## B2sv3               4.01e-01   1.38e-03  289.99  < 2e-16 ***
## error1              1.26e-01   1.37e-03   91.35  < 2e-16 ***
## error2              3.55e-01   1.38e-03  257.01  < 2e-16 ***
## ModelBeta2svTrueTRUE -9.63e-02   1.30e-03  -74.09  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.285 on 259186 degrees of freedom
## Multiple R-squared:  0.508,Adjusted R-squared:  0.508
## F-statistic: 2.06e+04 on 13 and 259186 DF,  p-value: <2e-16
```

Now add all three:

```
RMSEregression0 <- lm(B0RMSE ~ samplesize + metric + B0sv + B1sv + B2sv + error +
    ModelBeta0svTrue, data = rankData)
summary(RMSEregression0)
```

```
##
## Call:
## lm(formula = B0RMSE ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error + ModelBeta0svTrue, data = rankData)
##
## Residuals:
##     Min     1Q  Median     3Q     Max
## -0.6612 -0.1027 -0.0178  0.0727  2.6565
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.91e-01   1.48e-03  128.91  < 2e-16 ***
## samplesize          -2.87e-04   1.03e-06 -278.73  < 2e-16 ***
## metricGCV            4.23e-03   9.78e-04    4.33  1.5e-05 ***
## metricLOOCV          1.04e-02   9.78e-04   10.64  < 2e-16 ***
## metricSCV            1.82e-01   9.97e-04  182.76  < 2e-16 ***
## B0sv1                3.10e-01   8.58e-04  361.75  < 2e-16 ***
## B0sv3                6.75e-01   9.05e-04  745.99  < 2e-16 ***
## B1sv1                1.33e-02   8.47e-04   15.66  < 2e-16 ***
## B1sv3                6.20e-02   8.47e-04   73.18  < 2e-16 ***
## B2sv1                1.35e-02   8.47e-04   15.89  < 2e-16 ***
## B2sv3                6.36e-02   8.47e-04   75.08  < 2e-16 ***
## error1               1.09e-01   8.49e-04  128.32  < 2e-16 ***
## error2               2.99e-01   8.58e-04  348.72  < 2e-16 ***
## ModelBeta0svTrueTRUE -9.40e-02   9.96e-04  -94.43  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.176 on 259186 degrees of freedom
## Multiple R-squared:  0.774,Adjusted R-squared:  0.774
## F-statistic: 6.84e+04 on 13 and 259186 DF,  p-value: <2e-16
```

```
RMSEregression1 <- lm(B1RMSE ~ samplesize + metric + B0sv + B1sv + B2sv + error +
    ModelBeta1svTrue, data = rankData)
summary(RMSEregression1)
```

```
##
## Call:
## lm(formula = B1RMSE ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error + ModelBeta1svTrue, data = rankData)
##
## Residuals:
##    Min     1Q  Median     3Q     Max
## -0.820 -0.161 -0.026  0.113  5.368
##
```

```
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.60e-01   2.40e-03  108.37  < 2e-16 ***
## samplesize        -3.86e-04   1.66e-06 -232.97  < 2e-16 ***
## metricGCV          1.36e-02   1.57e-03    8.66  < 2e-16 ***
## metricLOOCV        2.37e-02   1.57e-03   15.02  < 2e-16 ***
## metricSCV          1.81e-01   1.57e-03  115.02  < 2e-16 ***
## B0sv1              1.04e-01   1.37e-03   75.39  < 2e-16 ***
## B0sv3              1.57e-01   1.41e-03  111.16  < 2e-16 ***
## B1sv1              9.77e-02   1.49e-03   65.51  < 2e-16 ***
## B1sv3              3.99e-01   1.38e-03  290.14  < 2e-16 ***
## B2sv1              5.86e-03   1.36e-03    4.30  1.7e-05 ***
## B2sv3              1.12e-02   1.36e-03    8.24  < 2e-16 ***
## error1             1.26e-01   1.37e-03   91.92  < 2e-16 ***
## error2             3.57e-01   1.37e-03  259.78  < 2e-16 ***
## ModelBeta1svTrueTRUE -9.59e-02 1.29e-03  -74.17  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.283 on 259186 degrees of freedom
## Multiple R-squared:  0.512,Adjusted R-squared:  0.512
## F-statistic: 2.09e+04 on 13 and 259186 DF,  p-value: <2e-16
```

```
RMSEregression2 <- lm(B2RMSE ~ samplesize + metric + B0sv + B1sv + B2sv + error +
    ModelBeta2svTrue, data = rankData)
summary(RMSEregression2)

##
## Call:
## lm(formula = B2RMSE ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error + ModelBeta2svTrue, data = rankData)
##
## Residuals:
##     Min     1Q Median     3Q    Max
## -0.857 -0.162 -0.026  0.113  4.462
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.60e-01   2.41e-03  107.81  < 2e-16 ***
## samplesize        -3.85e-04   1.67e-06 -231.09  < 2e-16 ***
## metricGCV          1.40e-02   1.58e-03    8.84  < 2e-16 ***
## metricLOOCV        2.45e-02   1.58e-03   15.47  < 2e-16 ***
## metricSCV          1.81e-01   1.58e-03  114.11  < 2e-16 ***
## B0sv1              1.00e-01   1.38e-03   72.30  < 2e-16 ***
## B0sv3              1.55e-01   1.42e-03  109.58  < 2e-16 ***
```

```
## B1sv1                8.18e-03   1.37e-03    5.96  2.5e-09 ***
## B1sv3                1.15e-02   1.37e-03    8.38  < 2e-16 ***
## B2sv1                9.82e-02   1.50e-03   65.45  < 2e-16 ***
## B2sv3                4.01e-01   1.38e-03  289.99  < 2e-16 ***
## error1               1.26e-01   1.37e-03   91.35  < 2e-16 ***
## error2               3.55e-01   1.38e-03  257.01  < 2e-16 ***
## ModelBeta2svTrueTRUE -9.63e-02   1.30e-03  -74.09  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.285 on 259186 degrees of freedom
## Multiple R-squared:  0.508,Adjusted R-squared:  0.508
## F-statistic: 2.06e+04 on 13 and 259186 DF,  p-value: <2e-16
```

3.2 RMSE Rank

How are the various parameters related to the results of the β_0 ranking?

```
rankData$B0rank = rankData[, "B0RMSE Rank"]
rankData$B1rank = rankData[, "B1RMSE Rank"]
rankData$B2rank = rankData[, "B2RMSE Rank"]
rankLM <- lm(B0rank ~ samplesize + metric + B0sv + B1sv + B2sv + error, data = rankData)
summary(rankLM)

##
## Call:
## lm(formula = B0rank ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error, data = rankData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -33.22  -6.29  -1.50   4.53  49.86
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.43e+01   7.25e-02  197.71  < 2e-16 ***
## samplesize  -5.53e-03   5.68e-05  -97.45  < 2e-16 ***
## metricGCV    3.52e-01   5.45e-02    6.46  1.0e-10 ***
## metricLOOCV  7.91e-01   5.45e-02   14.52  < 2e-16 ***
## metricSCV    1.23e+01   5.45e-02  225.74  < 2e-16 ***
## B0sv1       -6.07e+00   4.72e-02 -128.65  < 2e-16 ***
## B0sv3       -1.56e+01   4.72e-02 -330.65  < 2e-16 ***
## B1sv1        2.76e-01   4.72e-02    5.85  4.9e-09 ***
## B1sv3        1.10e+00   4.72e-02   23.40  < 2e-16 ***
## B2sv1        2.02e-01   4.72e-02    4.29  1.8e-05 ***
```

```
## B2sv3          1.16e+00   4.72e-02    24.65  < 2e-16 ***
## error1         3.06e+00   4.72e-02    64.94  < 2e-16 ***
## error2         6.88e+00   4.72e-02   145.79  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.8 on 259187 degrees of freedom
## Multiple R-squared:  0.454,Adjusted R-squared:  0.454
## F-statistic: 1.79e+04 on 12 and 259187 DF,  p-value: <2e-16
```

How are the various parameters related to the results of the β_1 ranking?

```
rankLM <- lm(B1rank ~ samplesize + metric + B0sv + B1sv + B2sv + error, data = rankData)
summary(rankLM)

##
## Call:
## lm(formula = B1rank ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error, data = rankData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -30.88  -9.35  -1.24   8.60  39.90
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.72e+00   8.56e-02  113.49  < 2e-16 ***
## samplesize  -2.15e-03   6.71e-05  -32.08  < 2e-16 ***
## metricGCV    5.64e-01   6.43e-02    8.77  < 2e-16 ***
## metricLOOCV  8.63e-01   6.43e-02   13.42  < 2e-16 ***
## metricSCV    8.83e+00   6.43e-02  137.25  < 2e-16 ***
## B0sv1        5.47e+00   5.57e-02   98.24  < 2e-16 ***
## B0sv3        6.01e-01   5.57e-02   10.79  < 2e-16 ***
## B1sv1        2.28e+00   5.57e-02   41.01  < 2e-16 ***
## B1sv3        1.66e+00   5.57e-02   29.76  < 2e-16 ***
## B2sv1        2.08e-01   5.57e-02    3.73  0.00019 ***
## B2sv3       -5.66e-01   5.57e-02  -10.16  < 2e-16 ***
## error1       3.30e+00   5.57e-02   59.25  < 2e-16 ***
## error2       5.59e+00   5.57e-02  100.34  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.6 on 259187 degrees of freedom
## Multiple R-squared:  0.163,Adjusted R-squared:  0.162
## F-statistic: 4.19e+03 on 12 and 259187 DF,  p-value: <2e-16
```

How are the various parameters related to the results of the β_2 ranking?

```
rankLM <- lm(B2rank ~ samplesize + metric + B0sv + B1sv + B2sv + error, data = rankData)
summary(rankLM)

##
## Call:
## lm(formula = B2rank ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error, data = rankData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -30.71  -9.35  -1.30   8.58  39.91
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.67e+00   8.57e-02  112.84  < 2e-16 ***
## samplesize  -2.21e-03   6.72e-05  -32.85  < 2e-16 ***
## metricGCV    5.85e-01   6.44e-02    9.08  < 2e-16 ***
## metricLOOCV  8.92e-01   6.44e-02   13.84  < 2e-16 ***
## metricSCV    8.80e+00   6.44e-02  136.70  < 2e-16 ***
## B0sv1        5.45e+00   5.58e-02   97.62  < 2e-16 ***
## B0sv3        5.65e-01   5.58e-02   10.13  < 2e-16 ***
## B1sv1        1.84e-01   5.58e-02    3.30  0.00098 ***
## B1sv3       -5.09e-01   5.58e-02   -9.12  < 2e-16 ***
## B2sv1        2.11e+00   5.58e-02   37.86  < 2e-16 ***
## B2sv3        1.72e+00   5.58e-02   30.77  < 2e-16 ***
## error1       3.50e+00   5.58e-02   62.75  < 2e-16 ***
## error2       5.71e+00   5.58e-02  102.32  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.6 on 259187 degrees of freedom
## Multiple R-squared:  0.162,Adjusted R-squared:  0.162
## F-statistic: 4.19e+03 on 12 and 259187 DF,  p-value: <2e-16
```

Now add in whether or not the correct model was specified.

```
rankLM <- lm(B0rank ~ samplesize + metric + B0sv + B1sv + B2sv + error + ModelBeta0svTrue,
    data = rankData)
summary(rankLM)

##
## Call:
## lm(formula = B0rank ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error + ModelBeta0svTrue, data = rankData)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -37.49  -5.23  -1.35   3.17  47.85
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.21e+01   7.60e-02  291.47  < 2e-16 ***
## samplesize           -3.89e-03   5.28e-05  -73.67  < 2e-16 ***
## metricGCV             4.14e-01   5.02e-02    8.25  < 2e-16 ***
## metricLOOCV           7.91e-01   5.02e-02   15.78  < 2e-16 ***
## metricSCV             1.01e+01   5.11e-02  198.49  < 2e-16 ***
## B0sv1                -4.56e+00   4.40e-02 -103.67  < 2e-16 ***
## B0sv3                -1.21e+01   4.64e-02 -260.60  < 2e-16 ***
## B1sv1                 2.59e-01   4.34e-02    5.96  2.5e-09 ***
## B1sv3                 9.69e-01   4.34e-02   22.30  < 2e-16 ***
## B2sv1                 1.85e-01   4.34e-02    4.27  2.0e-05 ***
## B2sv3                 1.02e+00   4.34e-02   23.49  < 2e-16 ***
## error1                2.44e+00   4.35e-02   55.97  < 2e-16 ***
## error2                5.38e+00   4.40e-02  122.22  < 2e-16 ***
## ModelBeta0svTrueTRUE -1.10e+01   5.11e-02 -215.42  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.03 on 259186 degrees of freedom
## Multiple R-squared:  0.537,Adjusted R-squared:  0.537
## F-statistic: 2.31e+04 on 13 and 259186 DF,  p-value: <2e-16
```

How are the various parameters related to the results of the β_1 ranking?

```
rankLM <- lm(B1rank ~ samplesize + metric + B0sv + B1sv + B2sv + error + ModelBeta1svTrue,
    data = rankData)
summary(rankLM)

##
## Call:
## lm(formula = B1rank ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error + ModelBeta1svTrue, data = rankData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -33.26  -8.49  -1.23   7.69  41.02
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          16.411754   0.094289  174.06  < 2e-16 ***
## samplesize           -0.000916   0.000065  -14.09  < 2e-16 ***
```

```
## metricGCV                0.647168   0.061838   10.47  < 2e-16 ***
## metricLOOCV              1.045673   0.061848   16.91  < 2e-16 ***
## metricSCV                8.914274   0.061838  144.16  < 2e-16 ***
## B0sv1                    4.516250   0.053952   83.71  < 2e-16 ***
## B0sv3                   -1.465153   0.055399  -26.45  < 2e-16 ***
## B1sv1                   -1.171852   0.058578  -20.01  < 2e-16 ***
## B1sv3                    0.590349   0.054050   10.92  < 2e-16 ***
## B2sv1                    0.177556   0.053551    3.32  0.00091 ***
## B2sv3                   -0.716090   0.053561  -13.37  < 2e-16 ***
## error1                   2.791083   0.053665   52.01  < 2e-16 ***
## error2                   4.639920   0.053946   86.01  < 2e-16 ***
## ModelBeta1svTrueTRUE -7.396007   0.050808 -145.57  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.1 on 259186 degrees of freedom
## Multiple R-squared:  0.226,Adjusted R-squared:  0.226
## F-statistic: 5.82e+03 on 13 and 259186 DF,  p-value: <2e-16
```

How are the various parameters related to the results of the β_2 ranking?

```
rankLM <- lm(B0rank ~ samplesize + metric + B0sv + B1sv + B2sv + error + ModelBeta2svTrue,
    data = rankData)
summary(rankLM)

##
## Call:
## lm(formula = B0rank ~ samplesize + metric + B0sv + B1sv + B2sv +
##     error + ModelBeta2svTrue, data = rankData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -34.22  -6.35  -1.50   4.63  50.57
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.57e+01   8.27e-02  189.52  < 2e-16 ***
## samplesize          -5.29e-03   5.71e-05  -92.52  < 2e-16 ***
## metricGCV            3.69e-01   5.43e-02    6.78  1.2e-11 ***
## metricLOOCV          8.28e-01   5.44e-02   15.23  < 2e-16 ***
## metricSCV            1.23e+01   5.43e-02  226.57  < 2e-16 ***
## B0sv1               -6.26e+00   4.74e-02 -132.07  < 2e-16 ***
## B0sv3               -1.60e+01   4.86e-02 -329.02  < 2e-16 ***
## B1sv1                2.77e-01   4.71e-02    5.89  3.9e-09 ***
## B1sv3                1.08e+00   4.71e-02   22.98  < 2e-16 ***
## B2sv1               -4.93e-01   5.15e-02   -9.58  < 2e-16 ***
```

```
## B2sv3                      9.58e-01    4.75e-02    20.18   < 2e-16 ***
## error1                     2.96e+00    4.72e-02    62.75   < 2e-16 ***
## error2                     6.69e+00    4.74e-02   141.06   < 2e-16 ***
## ModelBeta2svTrueTRUE -1.49e+00    4.46e-02   -33.40   < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.78 on 259186 degrees of freedom
## Multiple R-squared:  0.456,Adjusted R-squared:  0.456
## F-statistic: 1.67e+04 on 13 and 259186 DF,  p-value: <2e-16
```

Let's also throw in all three variables about whether the selected model has the same spatial variation as the true model for a certain beta.

```
rankLM <- lm(B0rank ~ samplesize + metric + B0sv + B1sv + B2sv + error + ModelBeta0svTrue +
    ModelBeta1svTrue + ModelBeta2svTrue, data = rankData)
summary(rankLM)

##
## Call:
## lm(formula = B0rank ~ samplesize + metric + B0sv + B1sv + B2sv +
##      error + ModelBeta0svTrue + ModelBeta1svTrue + ModelBeta2svTrue,
##      data = rankData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -37.89  -5.28  -1.42   3.30  46.96
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             2.46e+01    9.12e-02   269.37   < 2e-16 ***
## samplesize             -3.45e-03    5.34e-05   -64.53   < 2e-16 ***
## metricGCV               4.44e-01    4.99e-02     8.89   < 2e-16 ***
## metricLOOCV             8.59e-01    5.00e-02    17.19   < 2e-16 ***
## metricSCV               1.02e+01    5.09e-02   200.14   < 2e-16 ***
## B0sv1                  -4.92e+00    4.45e-02  -110.67   < 2e-16 ***
## B0sv3                  -1.29e+01    4.89e-02  -262.73   < 2e-16 ***
## B1sv1                  -3.71e-01    4.73e-02    -7.84   4.4e-15 ***
## B1sv3                   7.54e-01    4.37e-02    17.27   < 2e-16 ***
## B2sv1                  -4.54e-01    4.73e-02    -9.60   < 2e-16 ***
## B2sv3                   8.06e-01    4.36e-02    18.48   < 2e-16 ***
## error1                  2.25e+00    4.35e-02    51.75   < 2e-16 ***
## error2                  5.04e+00    4.44e-02   113.47   < 2e-16 ***
## ModelBeta0svTrueTRUE -1.10e+01    5.09e-02  -215.53   < 2e-16 ***
## ModelBeta1svTrueTRUE -1.35e+00    4.11e-02   -32.90   < 2e-16 ***
## ModelBeta2svTrueTRUE -1.36e+00    4.10e-02   -33.12   < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.99 on 259184 degrees of freedom
## Multiple R-squared:  0.541,Adjusted R-squared:  0.541
## F-statistic: 2.03e+04 on 15 and 259184 DF,  p-value: <2e-16
```

How are the various parameters related to the results of the β_1 ranking?

```
rankLM <- lm(B1rank ~ samplesize + metric + B0sv + B1sv + B2sv + error + ModelBeta0svTrue +
    ModelBeta1svTrue + ModelBeta2svTrue, data = rankData)
summary(rankLM)

##
## Call:
## lm(formula = B1rank ~ samplesize + metric + B0sv + B1sv + B2sv +
##      error + ModelBeta0svTrue + ModelBeta1svTrue + ModelBeta2svTrue,
##      data = rankData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -33.87  -8.33  -1.18   7.68  40.55
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.97e+01   1.12e-01  176.19  < 2e-16 ***
## samplesize           -2.33e-04   6.57e-05   -3.55  0.00039 ***
## metricGCV             6.76e-01   6.14e-02   11.02  < 2e-16 ***
## metricLOOCV           1.06e+00   6.14e-02   17.29  < 2e-16 ***
## metricSCV             8.16e+00   6.26e-02  130.49  < 2e-16 ***
## B0sv1                 4.97e+00   5.46e-02   90.96  < 2e-16 ***
## B0sv3                -3.95e-01   6.01e-02   -6.57  4.9e-11 ***
## B1sv1                -1.15e+00   5.81e-02  -19.73  < 2e-16 ***
## B1sv3                 5.42e-01   5.36e-02   10.10  < 2e-16 ***
## B2sv1                -1.49e-01   5.81e-02   -2.56  0.01053 *
## B2sv3                -8.60e-01   5.36e-02  -16.04  < 2e-16 ***
## error1                2.53e+00   5.35e-02   47.25  < 2e-16 ***
## error2                4.03e+00   5.45e-02   73.91  < 2e-16 ***
## ModelBeta0svTrueTRUE -3.89e+00   6.25e-02  -62.33  < 2e-16 ***
## ModelBeta1svTrueTRUE -7.33e+00   5.04e-02 -145.32  < 2e-16 ***
## ModelBeta2svTrueTRUE -6.87e-01   5.04e-02  -13.63  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11 on 259184 degrees of freedom
## Multiple R-squared:  0.238,Adjusted R-squared:  0.238
```

```
## F-statistic: 5.39e+03 on 15 and 259184 DF,  p-value: <2e-16
```

How are the various parameters related to the results of the β_2 ranking?

```
rankLM <- lm(B2rank ~ samplesize + metric + B0sv + B1sv + B2sv + error + ModelBeta0svTrue +
    ModelBeta1svTrue + ModelBeta2svTrue, data = rankData)
summary(rankLM)

##
## Call:
## lm(formula = B2rank ~ samplesize + metric + B0sv + B1sv + B2sv +
##      error + ModelBeta0svTrue + ModelBeta1svTrue + ModelBeta2svTrue,
##      data = rankData)
##
## Residuals:
##    Min    1Q Median    3Q    Max
## -32.17  -8.36  -1.21   7.65  41.48
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.98e+01   1.12e-01   177.02  < 2e-16 ***
## samplesize            -2.64e-04   6.57e-05    -4.02  5.8e-05 ***
## metricGCV              6.98e-01   6.14e-02    11.38  < 2e-16 ***
## metricLOOCV            1.10e+00   6.14e-02    17.88  < 2e-16 ***
## metricSCV              8.16e+00   6.26e-02   130.46  < 2e-16 ***
## B0sv1                  4.89e+00   5.46e-02    89.49  < 2e-16 ***
## B0sv3                 -4.91e-01   6.01e-02    -8.17  3.0e-16 ***
## B1sv1                 -1.77e-01   5.81e-02    -3.05   0.0023 **
## B1sv3                 -7.78e-01   5.36e-02   -14.50  < 2e-16 ***
## B2sv1                 -1.39e+00   5.81e-02   -23.92  < 2e-16 ***
## B2sv3                  6.21e-01   5.36e-02    11.58  < 2e-16 ***
## error1                 2.71e+00   5.35e-02    50.68  < 2e-16 ***
## error2                 4.13e+00   5.45e-02    75.79  < 2e-16 ***
## ModelBeta0svTrueTRUE  -3.85e+00   6.25e-02   -61.59  < 2e-16 ***
## ModelBeta1svTrueTRUE  -7.73e-01   5.04e-02   -15.32  < 2e-16 ***
## ModelBeta2svTrueTRUE  -7.48e+00   5.04e-02  -148.55  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11 on 259184 degrees of freedom
## Multiple R-squared:  0.24,Adjusted R-squared:  0.24
## F-statistic: 5.46e+03 on 15 and 259184 DF,  p-value: <2e-16
```

3.3 RMSE Rank Proportion

Let's convert ranks into a proportion and then use a logistic regression to look for patterns in rank vs. other variables like the metric and degree of spatial variation, etc.

How are the various parameters related to the results of the β_0 ranking?

```
rankData$B0rankP = rankData[, "B0RMSE Rank"]/50
rankData$B1rankP = rankData[, "B1RMSE Rank"]/50
rankData$B2rankP = rankData[, "B2RMSE Rank"]/50
mylogit <- glm(cbind(B0rankP, 1 - B0rankP) ~ samplesize + metric + B0sv + B1sv +
    B2sv + error, data = rankData, family = "binomial")

## Warning:  non-integer counts in a binomial glm!

summary(mylogit)

##
## Call:
## glm(formula = cbind(B0rankP, 1 - B0rankP) ~ samplesize + metric +
##     B0sv + B1sv + B2sv + error, family = "binomial", data = rankData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7108  -0.3114  -0.0971   0.1461   2.5635
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.14e+00   1.94e-02  -58.83  < 2e-16 ***
## samplesize  -7.64e-04   1.55e-05  -49.27  < 2e-16 ***
## metricGCV    5.47e-02   1.55e-02    3.53  0.00042 ***
## metricLOOCV  1.21e-01   1.54e-02    7.86  3.9e-15 ***
## metricSCV    1.47e+00   1.43e-02  102.15  < 2e-16 ***
## B0sv1       -6.37e-01   1.12e-02  -57.07  < 2e-16 ***
## B0sv3       -2.31e+00   1.58e-02 -146.08  < 2e-16 ***
## B1sv1        3.81e-02   1.26e-02    3.01  0.00258 **
## B1sv3        1.49e-01   1.25e-02   11.93  < 2e-16 ***
## B2sv1        2.80e-02   1.27e-02    2.21  0.02690 *
## B2sv3        1.57e-01   1.25e-02   12.56  < 2e-16 ***
## error1       4.54e-01   1.32e-02   34.44  < 2e-16 ***
## error2       9.30e-01   1.29e-02   72.37  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 101072  on 259199  degrees of freedom
```

26

```
## Residual deviance:  50829  on 259187  degrees of freedom
## AIC: 180384
##
## Number of Fisher Scoring iterations: 5
```

How are the various parameters related to the results of the β_1 ranking?

```
mylogit <- glm(cbind(B1rankP, 1 - B1rankP) ~ samplesize + metric + B0sv + B1sv +
    B2sv + error, data = rankData, family = "binomial")

## Warning:  non-integer counts in a binomial glm!

summary(mylogit)

##
## Call:
## glm(formula = cbind(B1rankP, 1 - B1rankP) ~ samplesize + metric +
##     B0sv + B1sv + B2sv + error, family = "binomial", data = rankData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3801  -0.4638  -0.0544   0.3592   1.7640
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.36e+00   1.64e-02  -82.87  < 2e-16 ***
## samplesize  -1.99e-04   1.25e-05  -15.96  < 2e-16 ***
## metricGCV    5.47e-02   1.22e-02    4.47  7.8e-06 ***
## metricLOOCV  8.33e-02   1.22e-02    6.82  8.9e-12 ***
## metricSCV    7.75e-01   1.18e-02   65.63  < 2e-16 ***
## B0sv1        4.95e-01   1.03e-02   48.05  < 2e-16 ***
## B0sv3        5.79e-02   1.06e-02    5.48  4.2e-08 ***
## B1sv1        2.12e-01   1.04e-02   20.43  < 2e-16 ***
## B1sv3        1.55e-01   1.04e-02   14.89  < 2e-16 ***
## B2sv1        1.91e-02   1.03e-02    1.85    0.064 .
## B2sv3       -5.25e-02   1.04e-02   -5.06  4.1e-07 ***
## error1       3.15e-01   1.05e-02   29.88  < 2e-16 ***
## error2       5.18e-01   1.04e-02   49.63  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 80156  on 259199  degrees of freedom
## Residual deviance: 68248  on 259187  degrees of freedom
## AIC: 279685
##
## Number of Fisher Scoring iterations: 4
```

How are the various parameters related to the results of the β_2 ranking?

```
mylogit <- glm(cbind(B2rankP, 1 - B2rankP) ~ samplesize + metric + B0sv + B1sv +
    B2sv + error, data = rankData, family = "binomial")
```

```
## Warning:  non-integer counts in a binomial glm!
```

```
summary(mylogit)
```

```
##
## Call:
## glm(formula = cbind(B2rankP, 1 - B2rankP) ~ samplesize + metric +
##     B0sv + B1sv + B2sv + error, family = "binomial", data = rankData)
##
## Deviance Residuals:
##     Min        1Q    Median        3Q       Max
## -1.3731   -0.4656   -0.0568    0.3582    1.7645
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.36e+00   1.64e-02  -83.12  < 2e-16 ***
## samplesize  -2.05e-04   1.25e-05  -16.36  < 2e-16 ***
## metricGCV    5.68e-02   1.22e-02    4.64  3.5e-06 ***
## metricLOOCV  8.61e-02   1.22e-02    7.05  1.8e-12 ***
## metricSCV    7.74e-01   1.18e-02   65.49  < 2e-16 ***
## B0sv1        4.92e-01   1.03e-02   47.82  < 2e-16 ***
## B0sv3        5.44e-02   1.06e-02    5.16  2.5e-07 ***
## B1sv1        1.69e-02   1.03e-02    1.64      0.1
## B1sv3       -4.72e-02   1.04e-02   -4.55  5.3e-06 ***
## B2sv1        1.96e-01   1.04e-02   18.91  < 2e-16 ***
## B2sv3        1.60e-01   1.04e-02   15.41  < 2e-16 ***
## error1       3.34e-01   1.05e-02   31.68  < 2e-16 ***
## error2       5.29e-01   1.04e-02   50.70  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 80411  on 259199  degrees of freedom
## Residual deviance: 68477  on 259187  degrees of freedom
## AIC: 279952
##
## Number of Fisher Scoring iterations: 4
```

Now add in whether the model selected a model with appropriate spatial variation for the parameter.

```
mylogit <- glm(cbind(B0rankP, 1 - B0rankP) ~ samplesize + metric + B0sv + B1sv +
    B2sv + error + ModelBeta0svTrue, data = rankData, family = "binomial")
```

Warning: non-integer counts in a binomial glm!

```
summary(mylogit)
```

```
##
## Call:
## glm(formula = cbind(B0rankP, 1 - B0rankP) ~ samplesize + metric +
##     B0sv + B1sv + B2sv + error + ModelBeta0svTrue, family = "binomial",
##     data = rankData)
##
## Deviance Residuals:
##    Min      1Q   Median      3Q      Max
## -1.855  -0.290  -0.107   0.118    2.424
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -5.13e-01   2.13e-02  -24.05  < 2e-16 ***
## samplesize           -6.11e-04   1.59e-05  -38.50  < 2e-16 ***
## metricGCV             6.22e-02   1.56e-02    3.98  6.9e-05 ***
## metricLOOCV           1.24e-01   1.55e-02    7.98  1.5e-15 ***
## metricSCV             1.28e+00   1.47e-02   86.69  < 2e-16 ***
## B0sv1                -5.08e-01   1.15e-02  -44.29  < 2e-16 ***
## B0sv3                -1.95e+00   1.65e-02 -118.05  < 2e-16 ***
## B1sv1                 3.63e-02   1.29e-02    2.82   0.0048 **
## B1sv3                 1.43e-01   1.27e-02   11.25  < 2e-16 ***
## B2sv1                 2.57e-02   1.29e-02    1.99   0.0463 *
## B2sv3                 1.51e-01   1.27e-02   11.85  < 2e-16 ***
## error1                4.01e-01   1.35e-02   29.68  < 2e-16 ***
## error2                8.04e-01   1.32e-02   60.94  < 2e-16 ***
## ModelBeta0svTrueTRUE -9.17e-01   1.23e-02  -74.77  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 101072  on 259199  degrees of freedom
## Residual deviance:  45313  on 259186  degrees of freedom
## AIC: 164507
##
## Number of Fisher Scoring iterations: 5
```

How are the various parameters related to the results of the β_1 ranking?

```
mylogit <- glm(cbind(B1rankP, 1 - B1rankP) ~ samplesize + metric + B0sv + B1sv +
    B2sv + error + ModelBeta1svTrue, data = rankData, family = "binomial")
```

Warning: non-integer counts in a binomial glm!

```
summary(mylogit)
```

```
##
## Call:
## glm(formula = cbind(B1rankP, 1 - B1rankP) ~ samplesize + metric +
##     B0sv + B1sv + B2sv + error + ModelBeta1svTrue, family = "binomial",
##     data = rankData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4851  -0.4292  -0.0549   0.3254   1.7851
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -7.89e-01   1.84e-02  -42.92  < 2e-16 ***
## samplesize           -9.38e-05   1.27e-05   -7.38  1.6e-13 ***
## metricGCV             6.34e-02   1.24e-02    5.13  2.9e-07 ***
## metricLOOCV           1.02e-01   1.23e-02    8.26  < 2e-16 ***
## metricSCV             7.99e-01   1.19e-02   66.96  < 2e-16 ***
## B0sv1                 4.20e-01   1.04e-02   40.23  < 2e-16 ***
## B0sv3                -1.26e-01   1.10e-02  -11.44  < 2e-16 ***
## B1sv1                -8.36e-02   1.14e-02   -7.36  1.9e-13 ***
## B1sv3                 7.04e-02   1.07e-02    6.61  3.8e-11 ***
## B2sv1                 1.65e-02   1.04e-02    1.58     0.11
## B2sv3                -6.71e-02   1.05e-02   -6.41  1.4e-10 ***
## error1                2.76e-01   1.07e-02   25.86  < 2e-16 ***
## error2                4.43e-01   1.06e-02   41.88  < 2e-16 ***
## ModelBeta1svTrueTRUE -6.67e-01   9.68e-03  -68.95  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 80156  on 259199  degrees of freedom
## Residual deviance: 63446  on 259186  degrees of freedom
## AIC: 268141
##
## Number of Fisher Scoring iterations: 4
```

How are the various parameters related to the results of the β_2 ranking?

```
mylogit <- glm(cbind(B2rankP, 1 - B2rankP) ~ samplesize + metric + B0sv + B1sv +
    B2sv + error + ModelBeta2svTrue, data = rankData, family = "binomial")

## Warning:  non-integer counts in a binomial glm!

summary(mylogit)

##
## Call:
## glm(formula = cbind(B2rankP, 1 - B2rankP) ~ samplesize + metric +
##      B0sv + B1sv + B2sv + error + ModelBeta2svTrue, family = "binomial",
##      data = rankData)
##
## Deviance Residuals:
##      Min       1Q    Median       3Q       Max
## -1.4716   -0.4292   -0.0559    0.3234    1.8502
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -7.88e-01   1.83e-02   -42.93   < 2e-16 ***
## samplesize            -9.81e-05   1.27e-05    -7.71   1.2e-14 ***
## metricGCV              6.56e-02   1.24e-02     5.30   1.1e-07 ***
## metricLOOCV            1.05e-01   1.23e-02     8.53   < 2e-16 ***
## metricSCV              7.99e-01   1.19e-02    66.91   < 2e-16 ***
## B0sv1                  4.15e-01   1.05e-02    39.72   < 2e-16 ***
## B0sv3                 -1.31e-01   1.10e-02   -11.88   < 2e-16 ***
## B1sv1                  1.76e-02   1.04e-02     1.69    0.092 .
## B1sv3                 -5.79e-02   1.05e-02    -5.54   3.1e-08 ***
## B2sv1                 -1.05e-01   1.14e-02    -9.25   < 2e-16 ***
## B2sv3                  7.85e-02   1.06e-02     7.38   1.6e-13 ***
## error1                 2.94e-01   1.07e-02    27.55   < 2e-16 ***
## error2                 4.54e-01   1.06e-02    42.89   < 2e-16 ***
## ModelBeta2svTrueTRUE  -6.81e-01   9.68e-03   -70.34   < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 80411  on 259199  degrees of freedom
## Residual deviance: 63478  on 259186  degrees of freedom
## AIC: 267742
##
## Number of Fisher Scoring iterations: 4
```

Let's also throw in all three variables about whether the selected model has the same spatial variation as the true model for a certain beta.

```
mylogit <- glm(cbind(B0rankP, 1 - B0rankP) ~ samplesize + metric + B0sv + B1sv +
    B2sv + error + ModelBeta0svTrue + ModelBeta1svTrue + ModelBeta2svTrue, data = rankData,
    family = "binomial")

## Warning:  non-integer counts in a binomial glm!

summary(mylogit)

##
## Call:
## glm(formula = cbind(B0rankP, 1 - B0rankP) ~ samplesize + metric +
##     B0sv + B1sv + B2sv + error + ModelBeta0svTrue + ModelBeta1svTrue +
##     ModelBeta2svTrue, family = "binomial", data = rankData)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.868  -0.289  -0.108   0.119   2.381
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.282751   0.024725  -11.44  < 2e-16 ***
## samplesize          -0.000572   0.000016  -35.65  < 2e-16 ***
## metricGCV            0.064103   0.015622    4.10  4.1e-05 ***
## metricLOOCV          0.129929   0.015499    8.38  < 2e-16 ***
## metricSCV            1.278847   0.014758   86.65  < 2e-16 ***
## B0sv1               -0.539710   0.011622  -46.44  < 2e-16 ***
## B0sv3               -2.028761   0.017087 -118.73  < 2e-16 ***
## B1sv1               -0.004352   0.013302   -0.33     0.74
## B1sv3                0.145155   0.012798   11.34  < 2e-16 ***
## B2sv1               -0.016051   0.013305   -1.21     0.23
## B2sv3                0.152720   0.012793   11.94  < 2e-16 ***
## error1               0.382086   0.013572   28.15  < 2e-16 ***
## error2               0.769845   0.013341   57.71  < 2e-16 ***
## ModelBeta0svTrueTRUE -0.913511   0.012313  -74.19  < 2e-16 ***
## ModelBeta1svTrueTRUE -0.148471   0.011513  -12.90  < 2e-16 ***
## ModelBeta2svTrueTRUE -0.149818   0.011507  -13.02  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 101072  on 259199  degrees of freedom
## Residual deviance:  44966  on 259184  degrees of freedom
## AIC: 163050
##
## Number of Fisher Scoring iterations: 5
```

How are the various parameters related to the results of the β_1 ranking?

```r
mylogit <- glm(cbind(B1rankP, 1 - B1rankP) ~ samplesize + metric + B0sv + B1sv +
    B2sv + error + ModelBeta0svTrue + ModelBeta1svTrue + ModelBeta2svTrue, data = rankData,
    family = "binomial")
```

```
## Warning:  non-integer counts in a binomial glm!
```

```r
summary(mylogit)
```

```
##
## Call:
## glm(formula = cbind(B1rankP, 1 - B1rankP) ~ samplesize + metric +
##     B0sv + B1sv + B2sv + error + ModelBeta0svTrue + ModelBeta1svTrue +
##     ModelBeta2svTrue, family = "binomial", data = rankData)
##
## Deviance Residuals:
##     Min       1Q    Median       3Q      Max
## -1.4815  -0.4238  -0.0539   0.3261   1.7835
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -4.90e-01   2.17e-02  -22.65  < 2e-16 ***
## samplesize             -2.68e-05   1.29e-05   -2.07    0.039 *
## metricGCV               6.70e-02   1.24e-02    5.40  6.6e-08 ***
## metricLOOCV             1.04e-01   1.24e-02    8.41  < 2e-16 ***
## metricSCV               7.35e-01   1.21e-02   60.62  < 2e-16 ***
## B0sv1                   4.77e-01   1.08e-02   44.23  < 2e-16 ***
## B0sv3                  -8.38e-03   1.22e-02   -0.69    0.491
## B1sv1                  -8.24e-02   1.14e-02   -7.24  4.5e-13 ***
## B1sv3                   6.39e-02   1.07e-02    5.99  2.1e-09 ***
## B2sv1                  -1.55e-02   1.13e-02   -1.37    0.170
## B2sv3                  -8.23e-02   1.06e-02   -7.79  6.7e-15 ***
## error1                  2.51e-01   1.07e-02   23.42  < 2e-16 ***
## error2                  3.86e-01   1.08e-02   35.80  < 2e-16 ***
## ModelBeta0svTrueTRUE   -3.63e-01   1.19e-02  -30.50  < 2e-16 ***
## ModelBeta1svTrueTRUE   -6.64e-01   9.70e-03  -68.50  < 2e-16 ***
## ModelBeta2svTrueTRUE   -6.93e-02   9.73e-03   -7.12  1.1e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 80156  on 259199  degrees of freedom
## Residual deviance: 62469  on 259184  degrees of freedom
## AIC: 266760
##
## Number of Fisher Scoring iterations: 4
```

How are the various parameters related to the results of the β_2 ranking?

```
mylogit <- glm(cbind(B2rankP, 1 - B2rankP) ~ samplesize + metric + B0sv + B1sv +
    B2sv + error + ModelBeta0svTrue + ModelBeta1svTrue + ModelBeta2svTrue, data = rankData,
    family = "binomial")

## Warning:  non-integer counts in a binomial glm!

summary(mylogit)

##
## Call:
## glm(formula = cbind(B2rankP, 1 - B2rankP) ~ samplesize + metric +
##     B0sv + B1sv + B2sv + error + ModelBeta0svTrue + ModelBeta1svTrue +
##     ModelBeta2svTrue, family = "binomial", data = rankData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4326  -0.4247  -0.0549   0.3241   1.8350
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.483274   0.021659  -22.31  < 2e-16 ***
## samplesize           -0.000030   0.000013   -2.32     0.02 *
## metricGCV             0.069224   0.012401    5.58  2.4e-08 ***
## metricLOOCV           0.107600   0.012376    8.69  < 2e-16 ***
## metricSCV             0.735940   0.012136   60.64  < 2e-16 ***
## B0sv1                 0.470095   0.010785   43.59  < 2e-16 ***
## B0sv3                -0.017409   0.012175   -1.43     0.15
## B1sv1                -0.017928   0.011325   -1.58     0.11
## B1sv3                -0.074452   0.010580   -7.04  2.0e-12 ***
## B2sv1                -0.105099   0.011397   -9.22  < 2e-16 ***
## B2sv3                 0.070950   0.010670    6.65  2.9e-11 ***
## error1                0.268943   0.010721   25.09  < 2e-16 ***
## error2                0.396269   0.010782   36.75  < 2e-16 ***
## ModelBeta0svTrueTRUE -0.358974   0.011910  -30.14  < 2e-16 ***
## ModelBeta1svTrueTRUE -0.077173   0.009735   -7.93  2.2e-15 ***
## ModelBeta2svTrueTRUE -0.678451   0.009695  -69.98  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 80411  on 259199  degrees of freedom
## Residual deviance: 62508  on 259184  degrees of freedom
## AIC: 266380
##
## Number of Fisher Scoring iterations: 4
```

3.4 Bandwidth Size

References

Chris Brunsdon, Stewart Fotheringham, and Martin Charlton. Geographically weighted regression-modelling spatial non-stationarity. *Journal of the Royal Statistical Society Series D The Statistician*, 47(3):431–443, 1998. ISSN 00390526. doi: 10.1111/1467-9884.00145. URL http://www.jstor.org/stable/2988625.

William S Cleveland and Susan J Devlin. Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting. *Journal of the American Statistical Association*, 83(403):596–610, June 1988. doi: 10.1080/01621459.1959. 10501996.

A. Stewart Fotheringham, Chris Brunsdon, and Martin Charlton. *Geographically Weighted Regression: the analysis of spatially varying relationships*. John Wiley & Sons, West Sussex, England, 2002.

Clive Loader. *Local Regression and Likelihood*. Springer-Verlag, New York, NY, 1999.

Daniel P. McMillen. Perspectives on Spatial Econometrics: Linear Smoothing With Structured Models. *Journal of Regional Science*, 52(2):192–209, May 2012. ISSN 00224146. doi: 10.1111/j.1467-9787.2011.00746.x. URL http://doi.wiley.com/10.1111/j.1467-9787.2011.00746.x.

Daniel P. McMillen and Christian L. Redfearn. Estimation and Hypothesis Testing for Nonparametric Hedonic House Price Functions. *Journal of Regional Science*, 50(3):712–733, April 2010. ISSN 00224146. doi: 10.1111/j.1467-9787.2010.00664.x. URL http://doi.wiley.com/10.1111/j.1467-9787.2010.00664.x.

Antonio Paez, Steven Farber, and David Wheeler. A simulation-based study of geographically weighted regression as a method for investigating spatially varying relationships. *Environment and Planning A*, 43(12):2992–3010, 2011.