Software Estimation with Bayesian Regression

Chris Davey

10/23/2016

Overview

This document summarises the procedure of applying linear regression to data collected from a variety of software projects in order to construct a bayesian model for linear regression so as to predict a likely region of hours for a given project and a second model used to predict the likelihood of defects for a given project.

This document provides an illustration of the process, and it is possible after having acquired further data, to update the model, or introduce additional predictors and evaluate their contribution to the model as well as perform subsequent evaluation of the model against earlier iterations using measures such as information criteria.

Note however the data set available is limited, so this example is provided only as an example of method, additionally the attributes for the data set is also relatively limited, and additional attributes would be worth seeking out depending on data collection capability, if performing a similar task.

There is some considerations to be given to collecting data from multiple sources for software estimation, and these are summarised in the Kocaguneli et al article "When to use Data from other projects for effort estimation" along with an approach to use a distance based weighting determining which examples should be included [1].

The data set that was used in this example was acquired from several resources, these files are located in the **data** directory are are:

- cosmic.arff and isbsg10.arff

Acquired from the Tera data pages offering additional links to research on empirical methods applied to software engineering processes. The data sets are available at the cosmic data set page and the isbg data set page.

This data is a teaser data set and many of the attributes of interest were undefined, only a subset of the data was used. However the same procedure can be applied to the full data set (available commercially from the ISBSG site).
The original arff files and the converted csv files for this data is checked in with the data directory.

- project_tom_data.csv

The additional data is collected manually from example projects in my role at work. This includes additional information as to number of logical components in each project, as well as test case count. Only the numeric values useful for the analysis are retained, individual client names or project codes have been removed. The source data is ignored from the check-in, and is mixed into the other two data sets, the manual data cannot be identified from the merged data set.

Both groups of data have only a few common attributes, they were merged togethor by selecting a set of attributes that reflect the following data points.

- **functionPoints**

The function point estimate, in the case of the cosmic and isbg data sets these are produced using one of the function point estimation methods nominated in the data set. In the case of the data set collected manually, this reflects either the number of requirements or number of key test cases. The manual data reflects those features of the project that are significant in having been identified in the requirement set. The other data sets may have used a more formal approach. However, the number of function points (and indirectly requirements) has a positive correlation with the number of hours and defects that result in the project as shown in the correlation diagram.

For the Tera data sets this was derived from the FS property, in the manual data set this is the requirementCnt property.

- **teamSize**

The maximum size of the team working on the project. The assumption that the larger the team size, there is increased effort in communication about the intent, design, organisation of work, and testing, and defect fixing to name a few concerns. This will also result in increased overall effort for the project.

For the Tera data sets this was the MTS property, for the manual data set this was the teamSize property.

- **totalDefects**

Total defects is a candidate target variable, however this data set does not contain other attributes beyond function points, team size and estimatedHrs (and most of the estimated hours are unavailable). It would be worth while gathering more data to build into a model to predict the potential for defects in the project. Note, as a rule of thumb, the test team at work have held that roughly 13% of test cases may potentially fail resulting in one or more defects, and projects that start to exhibit higher defect rates generally indicate special attention is required. This is more of a conventional belief that has arisen out of experience with a large number of projects, it is based on a reasonable practice of collating test execution records, and has some amount of practical truth to it. However, it would be worth while putting this assumption to the test, and gathering more metrics around size of the project, number of unit tests, coverage and other details in order to model the risk of defect rates and determine whether quality assurance can be improved with efforts in areas such as unit testing, code review and integration testing, which may be assisted by a quicker develop, deploy and test cycle where the development team are more empowered by a quicker means of obtaining feed back and collecting details on that feedback.

The Tera data set defined multiple properties for defects, and this was obtained as the combination of "TOT_Defects", "TOT_B_Defects" and "TOT_I_Defects". For the manual data set this was derived from the property for "totalJiraDefects".

- **estimatedHrs**

Where available the data for estimated hours is included, however where it is not available it has been set to 0.

For the Tera cosmic data set this was a combination of the properties, "E_Plan", "E_Design", "E_Specify", "E_Build", "E_Implement", "E_Test". For the isbsg data set this was the "E_Estimate" value. And for the manual data this was the "projBudgetHrs". All estimates are in hours.

- **recordedHrs**

The actual number of hours recorded for the project.

For the Tera data set this was derived from the "N_effort" attribute, and for the manual data set this was derived from the "totalBudgetHrs" attribute.

Only complete cases of interest are maintained in the merged data set, which unfortunately reduces the size of the merged data set significantly.

However, it would be possible to merge additional data sets should they also contain suitable attributes that would map into the properties above as a minimal analsys. Further analysis is also worth pursuing if additional attributes were available, and common accross multiple data sets.

Note that there is some speculation that having data available from multiple organisations in aide of effort estimation may not be as an effective means of estimation as having good records from within the same company performing the estimation (refer to Deng Kefu). Hence good data collection on project size, quality metrics, effort estimates and actual effort would prove valuable for estimation purposes in the long run. As this data would reflect the internal processes and practices held by the company undertaking the collection.

This article demonstrates some methods for analysing this data, unfortunately the conclusions drawn with this limited data set are merely speculative. However, the techniques can be applied where there is sufficient data available.

Summary of Available Data

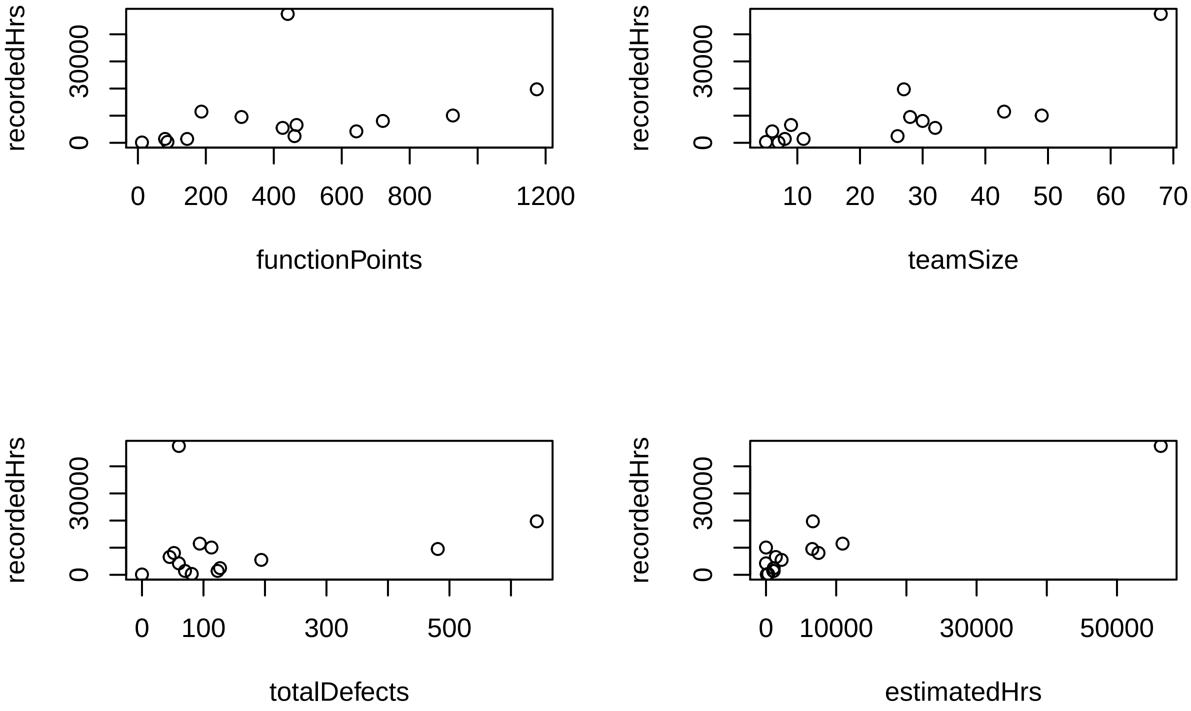
The resulting data set contains the number of samples below:

```
## [1] 14
```

Each attribute has the following summary

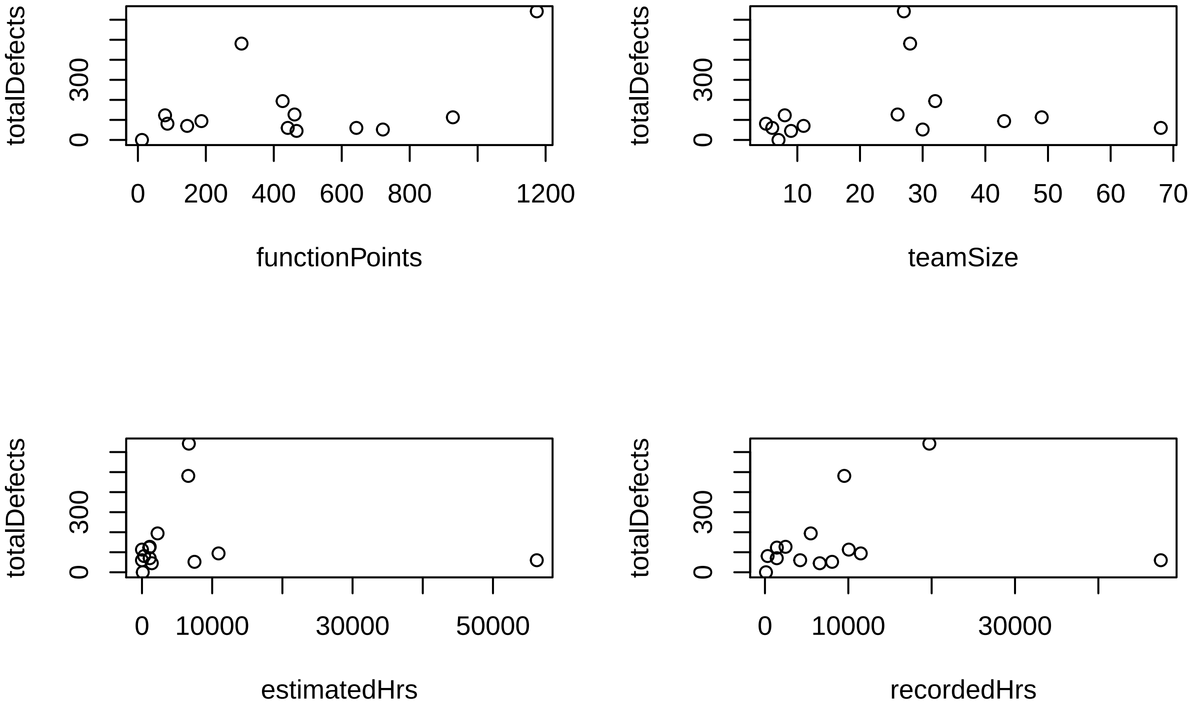
```
##  functionPoints      teamSize      totalDefects    estimatedHrs
##  Min.   :  12.0   Min.   : 5.00   Min.   :  0.0   Min.   :    0.0
##  1st Qu.: 155.5   1st Qu.: 8.25   1st Qu.: 60.0   1st Qu.:  490.6
##  Median : 433.5   Median :26.50   Median : 87.5   Median : 1254.3
##  Mean   : 434.0   Mean   :24.93   Mean   :153.0   Mean   : 6800.3
##  3rd Qu.: 599.0   3rd Qu.:31.50   3rd Qu.:126.0   3rd Qu.: 6660.0
##  Max.   :1174.0   Max.   :68.00   Max.   :642.0   Max.   :56239.0
##   recordedHrs
##  Min.   :  121.8
##  1st Qu.: 1689.4
##  Median : 6032.4
##  Mean   : 9170.7
##  3rd Qu.: 9922.9
##  Max.   :47493.0
```

The following set of scatter plots give an indication of the relation between the target column “recordedHrs” and the other attributes.



In general there is a positive relationship between the target variable recorded hours and the other attributes.

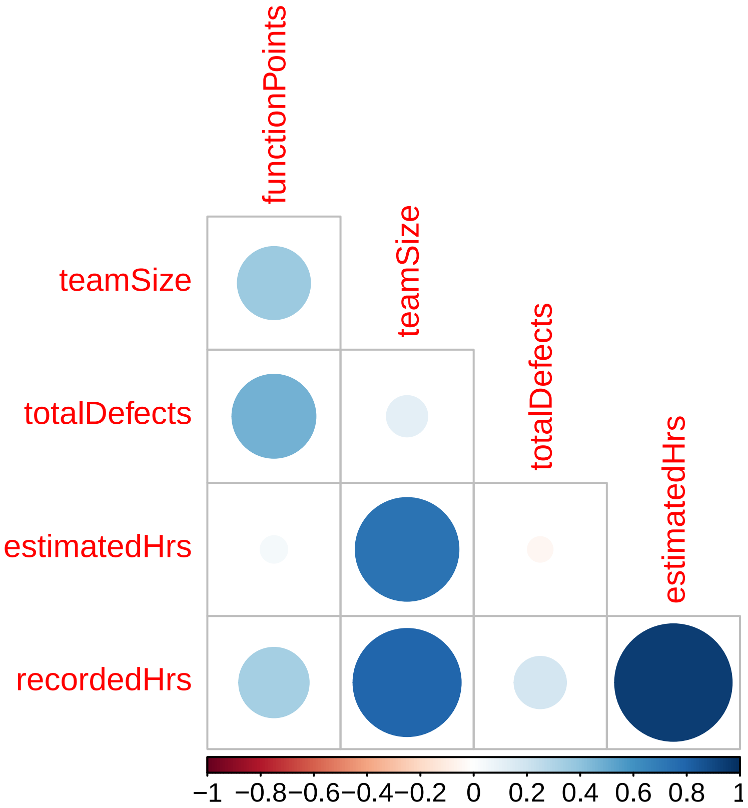
The scatter plots against totalDefects are also shown below.



The relation between totalDefects and the other attributes may suggest a postive relationship while at the same time, there does appear to be some unusual extreme values visible, especially in the far right hand corner.

While both scatter plots demonstrate that there is a lack of data (only a few sample points), it would be useful to acquire or gather additional information (either by purchasing data, or implementing a data collection method for software project metrics where possible within the organisation).

The same attributes have the following correlations plotted below



The correlation for each of the attributes is listed below

```
##                functionPoints  teamSize totalDefects estimatedHrs
## functionPoints      1.0000000 0.3617090   0.47884095   0.04873320
## teamSize            0.3617090 1.0000000   0.11299854   0.73342379
## totalDefects        0.4788410 0.1129985   1.00000000  -0.04189483
## estimatedHrs        0.0487332 0.7334238  -0.04189483   1.00000000
## recordedHrs         0.3350098 0.7979481   0.18314746   0.94388226
##                recordedHrs
## functionPoints   0.3350098
## teamSize         0.7979481
## totalDefects     0.1831475
## estimatedHrs     0.9438823
## recordedHrs      1.0000000
```

Perhaps unsurprisingly there is a positive correlation between most attributes, but most significantly between teamSize and estimatedHrs, and teamSize and recordedHrs. Perhaps surprisingly there is a negative correlation between totalDefects and estimatedHrs, this may be due to estimates tending to be overly optimistic.

Under a t-test for a 95 % confidence interval we seek a p-value less than :

```
## [1] 0.025
```

in order to reject the null hypothesis that the attributes are uncorrelated. Note in the case where two attributes are correlated this would suggest that they may contain a linear dependence and the regression may benefit from having one of the attributes removed. However for the sake of this example, we will not remove the highly correlated attributes.

The correlation between teamSize and recordedHrs has the p-value

```
## [1] 0.0006257728
```

and is flagged as indicating that the correlation with recordedHrs is likely

```
## [1] TRUE
```

The same procedure can be repeated for the other attributes against the recordedHrs attribute in order to determine whether the correlation is feasible.

```
## [1] "T-Test for correlation of recordedHrs to other attributes"

##       attributes         test  flag
## 1 functionPoints 2.416596e-01 FALSE
## 2       teamSize 6.257728e-04  TRUE
## 3   totalDefects 5.308417e-01 FALSE
## 4   estimatedHrs 3.992871e-07  TRUE
```

Interestingly, the functionPoints, and totalDefects from the recorded data do not test in such a manner as to suggest it correlates with recordedHrs, whereas the correlation by itself suggested this relation was positive.

The same procedure can be applied to gather some indication of relationship between the attributes and the second target variable, totalDefects.

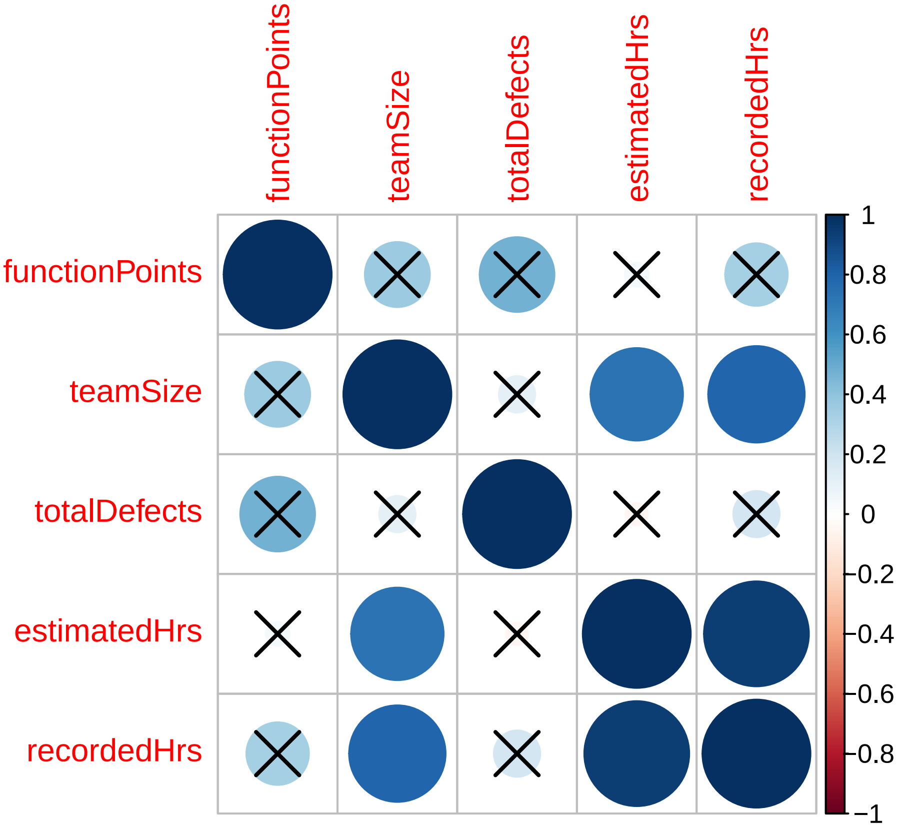
```
## [1] "T-Test for correlation of totalDefects to other attributes"

##       attributes       test  flag
## 1 functionPoints 0.08322821 FALSE
## 2       teamSize 0.70052028 FALSE
## 3   estimatedHrs 0.88692045 FALSE
## 4    recordedHrs 0.53084175 FALSE
```

Which interestingly suggests the totalDefects is not siginificantly correlated with any other attributes, hence may be independent of the other attributes. This independence would identify it as a potentially useful predictor when it is not used as a target variable.

The same method can be expanded to give a summary of linear dependence between the features using a test for significance between MxM features.

```
## [1] "p-values for test of correlation of attributes"

##                functionPoints     teamSize totalDefects estimatedHrs
## functionPoints     0.00000000 0.2038087796   0.08322821 8.685984e-01
## teamSize           0.20380878 0.0000000000   0.70052028 2.834806e-03
## totalDefects       0.08322821 0.7005202760   0.00000000 8.869204e-01
## estimatedHrs       0.86859839 0.0028348060   0.88692045 0.000000e+00
## recordedHrs        0.24165964 0.0006257728   0.53084175 3.992871e-07
##                 recordedHrs
## functionPoints 2.416596e-01
## teamSize       6.257728e-04
## totalDefects   5.308417e-01
## estimatedHrs   3.992871e-07
## recordedHrs    0.000000e+00
## [1] "T-Test for test of correlation of attributes"
```

```
##                functionPoints teamSize totalDefects estimatedHrs
## functionPoints           TRUE    FALSE        FALSE        FALSE
## teamSize                FALSE     TRUE        FALSE         TRUE
## totalDefects            FALSE    FALSE         TRUE        FALSE
## estimatedHrs            FALSE     TRUE        FALSE         TRUE
## recordedHrs             FALSE     TRUE        FALSE         TRUE
##                recordedHrs
## functionPoints       FALSE
## teamSize              TRUE
## totalDefects         FALSE
## estimatedHrs          TRUE
## recordedHrs           TRUE
```



This process is a kind of feature selection, the aim is to remove redundant predictors from the attributes, where those attributes are correlated.

Note that estimatedHrs and recordedHrs are strongly correlated and the test confirms this correlation exists, additionally teamSize is positively correlated with both estimatedHrs and recordedHrs and the test indicates this correlation exists. Hence, there is a linear dependence between these three attributes. When estimatedHrs and teamSize are used as predictors it may be recommended to remove one of these attributes from the predictor columns.

Note also that the method of testing is a point estimate approach, later on when we examine parameter space confidence intervals will be more useful. However so far this approach gives some indication as to the kind of relationship that exists within the data set, and some brief idea as to the ranges available for the values within the data set.

Modelling Prior and Posterior Distributions

Drawing on the example data available, we will make use of bayes least squares regression. However before applying this to the model, some description of the process being applied in the model is required.

The approach uses a linear regression, given the data we assume that the distribution of Y, the predictors X and the parameters β and σ^2 are distributed as

$$p(Y|\beta, \sigma^2, X) \sim N(X\beta, \sigma^2 I)$$

we are assuming that Y has a normal distribution about the product of X and β with a uniform variance σ^2 (I is the diagonal identity matrix). The conditional distribution of the parameters is assumed to be uniform

$$p(\beta, \sigma^2|X) \propto \sigma^{-2}$$

The posterior distribution of the parameters are given by the rule

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

The posterior distribution of the parameters becomes

$$p(\beta, \sigma^2|Y, X) \propto N(X\beta, \sigma^2 I) \times \sigma^{-2}$$

And factors into the product of the multivariate normal distribution and the inverse chi squared distribution

$$\propto N\left(\hat{\beta}, \sigma^2(X'X)^{-1}\right) \times \text{Inverse-}\chi^2(\nu, s^2)$$

where the estimate of β is

$$\hat{\beta} = (X'X)^{-1}X'Y$$

ν is the degrees of freedom

$$\nu = n - k$$

Note that X is of dimension $n \times k$ and that $n \geq k$. The estimate of the variance is derived from:

$$s^2 = (1/\nu)(Y - \bar{Y})'(Y - \bar{Y})$$

$$\bar{Y} = X\hat{\beta}$$

The posterior predictive distribution of new target variable $\tilde{Y}$ given new predictor examples $\tilde{X}$ and the posterior for parameters results as the product of the likelihood of $\tilde{Y}$ given the new data and the posterior distribution of the parameters from the previous data.

$$p(\tilde{Y}|\beta, \sigma^2, \tilde{X}, Y, X) = f(\tilde{Y}|\beta, \sigma^2, \tilde{X}) \times p(\beta, \sigma^2|Y, X)$$

After marginalising the predictive density $p(\tilde{Y}|Y, \tilde{X}, X)$ becomes a multivariate t-distribution centered at $\tilde{X}\hat{\beta}$, scaled by $[I + \tilde{X}V\tilde{X}']s^2$ with $\nu = n - k$ degress of freedom. Note that $V = (X'X)^{-1}$ and $\hat{\beta} = (X'X)^{-1}X'Y = VX'Y$

Update Rules

When presented with new observations it is also possible to update the distribution of the parameters for β and σ using the following update rules

$$s_n^2 = s^2 + \frac{1}{n_{new} - k}(Y_{new} - X_{new}\hat{\beta})'(Y_{new} - X_{new}\hat{\beta})$$

and the new degrees of freedom

$$\nu_n = \nu + (n_{new} - k)$$

which yields the parameters for the Inverse-$\chi^2(\nu_n, s_n^2)$ distribution. The β parameter is updated as

$$\mu_0 = \hat{\beta} = V_\beta X'Y$$

$$\Lambda_0 = V_\beta$$

$$\Lambda_n = \Lambda_0^{-1} + \Sigma^{-1}$$

$$u_n = (\Lambda_0^{-1} + \Sigma^{-1})^{-1}(\Lambda_0^{-1}\mu_0 + \Sigma^{-1}\bar{Y}_{new})$$

and $\Sigma = X'_{new}X_{new}$. The updated distribution for β becomes

$$p(\beta|\sigma^2, Y_{new}, Y, X_{new}, X) \sim N(\mu_n, \sigma^2\Lambda_n)$$

and the updated joint posterior distribution becomes

$$p(\beta, \sigma^2|Y_{new}, Y, X_{new}, X) \propto N(\mu_n, \sigma^2\Lambda_n) \times \text{Inverse-}\chi^2(\nu_n, s_n^2)$$

The new parameter estimates μ_n and $\sigma^2\Lambda_n$, ν_n and s_n^2 can be substituted into the regression model in order to update the model estimates, as well as used within the predictive distribution. Ongoing model updates can occur in the same procedure.

The initial effect of the uninformative prior is equivalent to standard linear regression. However one of the advantages of generating the bayesian model is the ability to perform an update of the parameters when new data becomes available. Details and further references are discussed here at Notes on Bayes Linear Regression as well as definitive resources to be found in Box and Tiao [2] and Gelman [3].

Inspecting Boundaries of Predictors with Credible intervals and the Highest Posterior Density

Once the model is trained we can inspect the upper and lower bounds of the parameter space, mainly the upper and lower bounds of the estimated mean of the predictor variables.

The boundary is determined on the marginal posterior distribution of the β parameter, which after marginalisation is a multivariate t-distribution centered at $\hat{\beta}$ and scaled by $X'Xs^2$ with $\nu = n - k$ degrees of freedom.

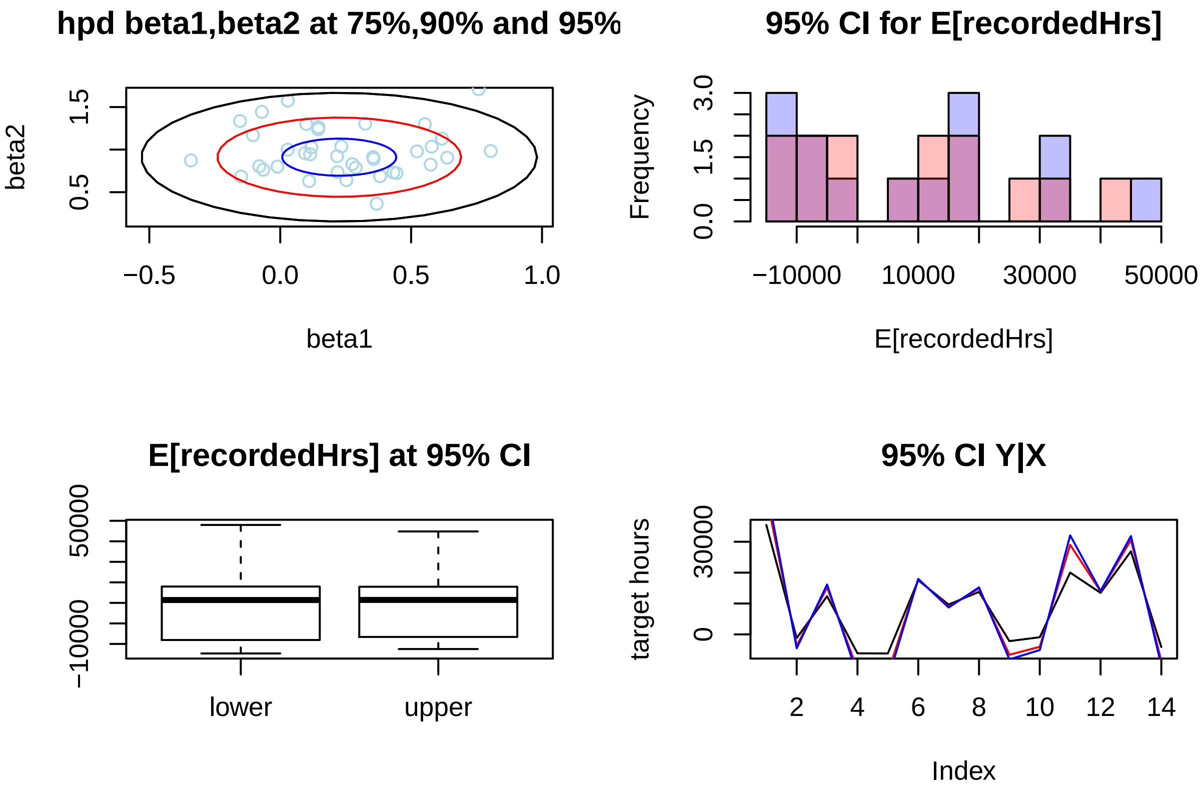
$$p(\beta|Y, X) \propto \left[1 + \frac{(\beta - \hat{\beta})'X'X(\beta - \hat{\beta})}{\nu s^2}\right]^{-\frac{1}{2}(\nu+k)}$$

Using the posterior distributions it is then possible to obtain credible intervals for the parameters at a given α level. For the parameter β this can be achieved via the multivariate t-distribution $p(\beta|Y, X)$ when the variance is assumed known we can make use of the t values for the $1 - \alpha$ level

$$\hat{\beta} \pm t_{\nu,\alpha/2}\sqrt{(X'X)s^2}$$

The first plot examines the first two dimensions of the parameter space of the β parameter matrix, the other plots examine the confidence intervals for the target variable recordedHrs based on the model at the 95% confidence interval. Note that under the normal distribution the sign of the extremum can be negative, although in the case of the project durations, this value does not realistically apply.

```r
## columns 1:4 will be the predictors
xcols <- 1:4
# target column is 5
ycol <- c(5)
test <- bayes_least_squares_uninform_prior(data, xcols, ycol, zscale=TRUE)
```

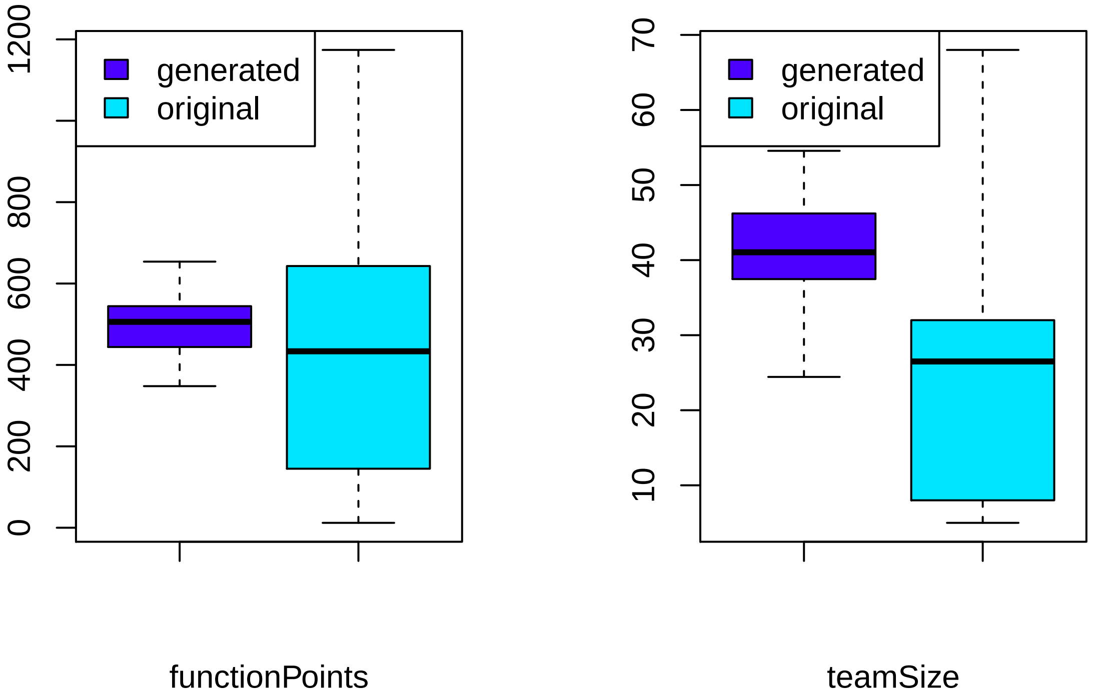


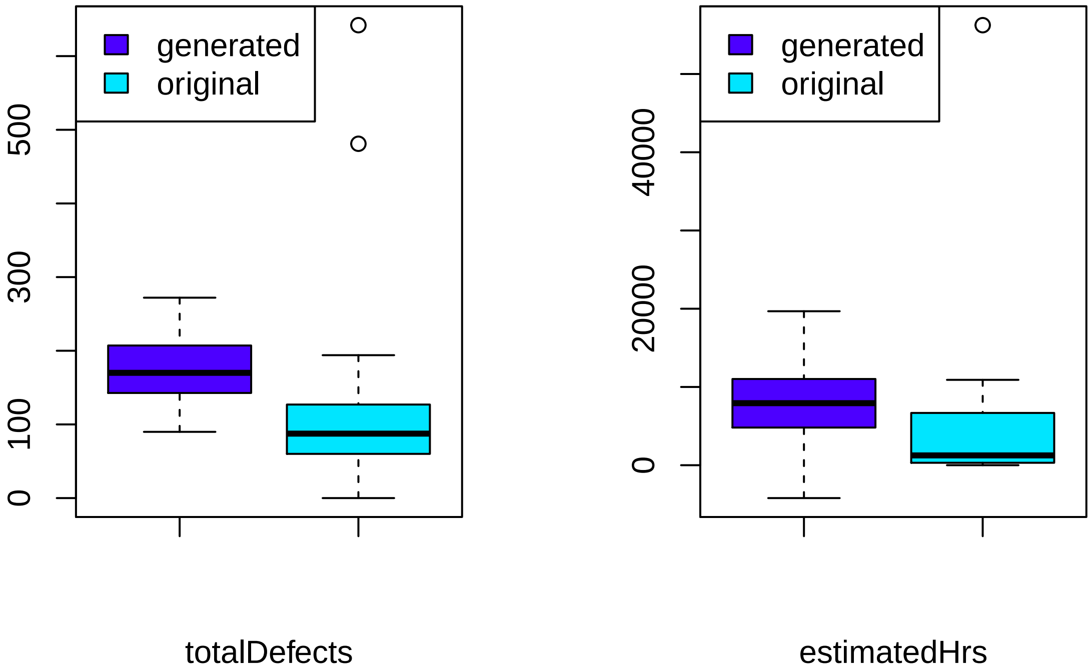
The upper confidence interval has a slighly higher distribution than the lower confidence interval. The summary of each being

```
## [1] "Summary of upper 95% CI"

##          V1
##  Min.   :-12498
##  1st Qu.: -5952
##  Median : 11433
##  Mean   :  9171
##  3rd Qu.: 17192
##  Max.   : 44811

## [1] "Summary of lower 95% CI"

##          V1
##  Min.   :-14641
##  1st Qu.: -7335
##  Median : 11392
##  Mean   :  9171
##  3rd Qu.: 17518
##  Max.   : 47983
```

Using the HPD of the beta interval is it also possible to gauge some idea of the distribution of the mean for the predictors this is derived by sampling from the posterior distribution for the beta-parameter, by doing so we can obtain expected values for the mean of the predictors by transforming the beta parameters back into the predictor parameters (this is done via scaling and offsetting, which have been stored during the training procedure). It gives an indication of the distribution that the model has defined for the expected values of the predictors.

```
len <- 100
d2 <- test$betaConfData(test, test$sSqr,len, names= c("B1", "B2", "B3", "B4", "prob"))

rConf = data.frame(b1=d2[,1]*test$xscale[1] + test$xcenter[1],
                   b2=d2[,2]*test$xscale[2] + test$xcenter[2],
                   b3=d2[,3]*test$xscale[3] + test$xcenter[3],
                   b4=d2[,4]*test$xscale[4] + test$xcenter[4])
colnames(rConf) <- colnames(data)[1:4]
```





totalDefects

estimatedHrs

The summary of the confidence intervals for the mean of the predicted values are below, we have two different distributions for the predictor variables, that of the original data source, and that which is generated based on the expected values for the parameter of the model. We can additionally form upper and lower confidence thresholds for predictor values in the same manner.

```
## [1] "Summary of predictor attributes from source data"
```

```
##  functionPoints      teamSize      totalDefects    estimatedHrs
##  Min.   :  12.0   Min.   : 5.00   Min.   :  0.0   Min.   :    0.0
##  1st Qu.: 155.5   1st Qu.: 8.25   1st Qu.: 60.0   1st Qu.:  490.6
##  Median : 433.5   Median :26.50   Median : 87.5   Median : 1254.3
##  Mean   : 434.0   Mean   :24.93   Mean   :153.0   Mean   : 6800.3
##  3rd Qu.: 599.0   3rd Qu.:31.50   3rd Qu.:126.0   3rd Qu.: 6660.0
##  Max.   :1174.0   Max.   :68.00   Max.   :642.0   Max.   :56239.0
```

```
## [1] "Summary of expected values for predictor attributes"
```

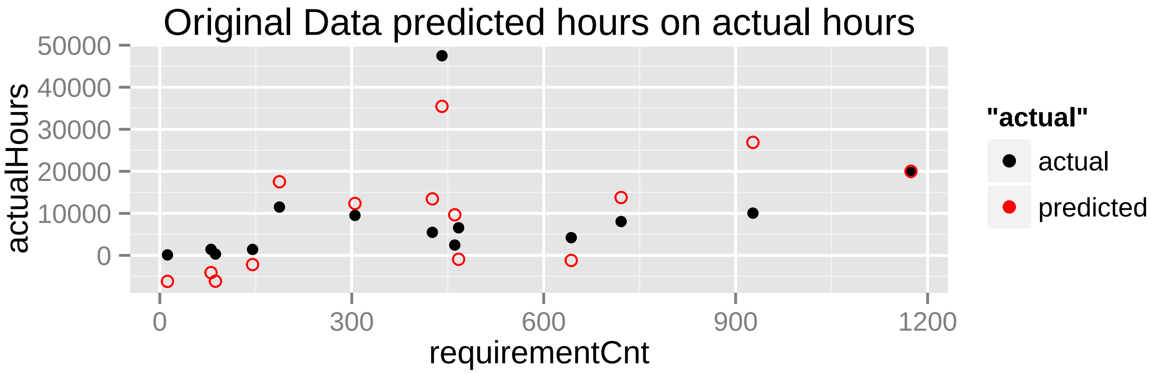
```
##  functionPoints     teamSize      totalDefects     estimatedHrs
##  Min.   :347.9   Min.   :24.44   Min.   : 89.92   Min.   :-4202
##  1st Qu.:444.3   1st Qu.:37.54   1st Qu.:143.14   1st Qu.: 4834
##  Median :505.8   Median :41.04   Median :170.04   Median : 7926
##  Mean   :497.6   Mean   :41.50   Mean   :175.37   Mean   : 8091
##  3rd Qu.:543.9   3rd Qu.:46.17   3rd Qu.:206.87   3rd Qu.:10954
##  Max.   :653.8   Max.   :54.56   Max.   :272.05   Max.   :19679
```

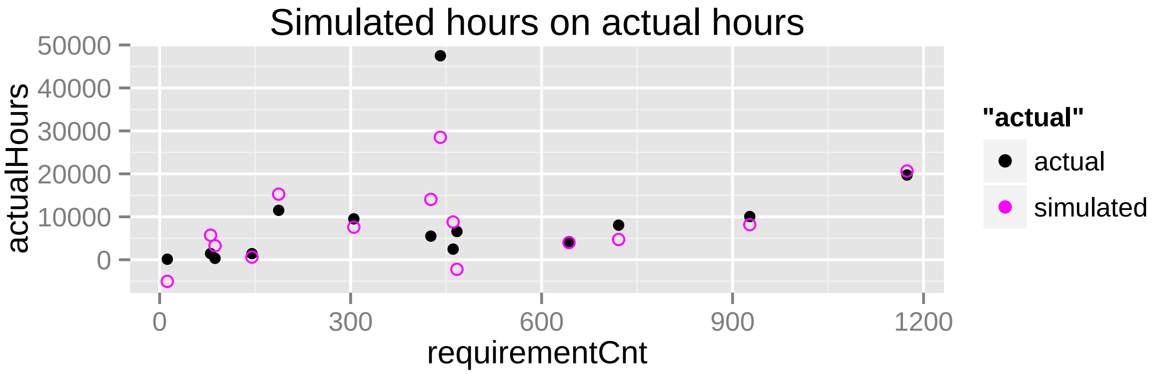
This also provides some insight into how the model may be used in a generative manner, where sampling form the parameter space can be used for example in simulation, or generation of representative data sets for the predictor variables, based on the posterior distribution of the parameters.

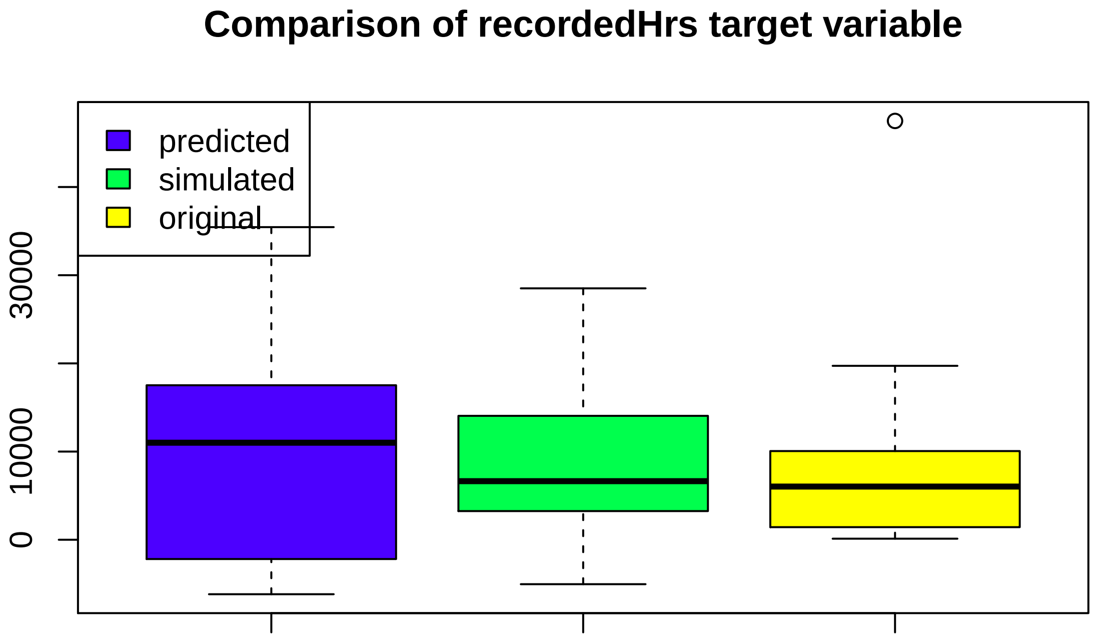
Application to prediction and simulation

Additionally it is possible to demonstrate the use of prediction of the model using the parameters derived from the non-informative prior. The prediction of the current data set is shown in the first plot, and a simulation of the predictive posterior generated from the multivariate normal distribution from the posterior parameters is shown below.

```
len <- 1000
X1 <- data[,xcols]
Y1 <- data[,ycol]
## investigate the prediction
Ypred <- test$predictY(test, X1)
Ypred1 <- test$yPredictivePosteriorSimulate(test,X1, len)
```



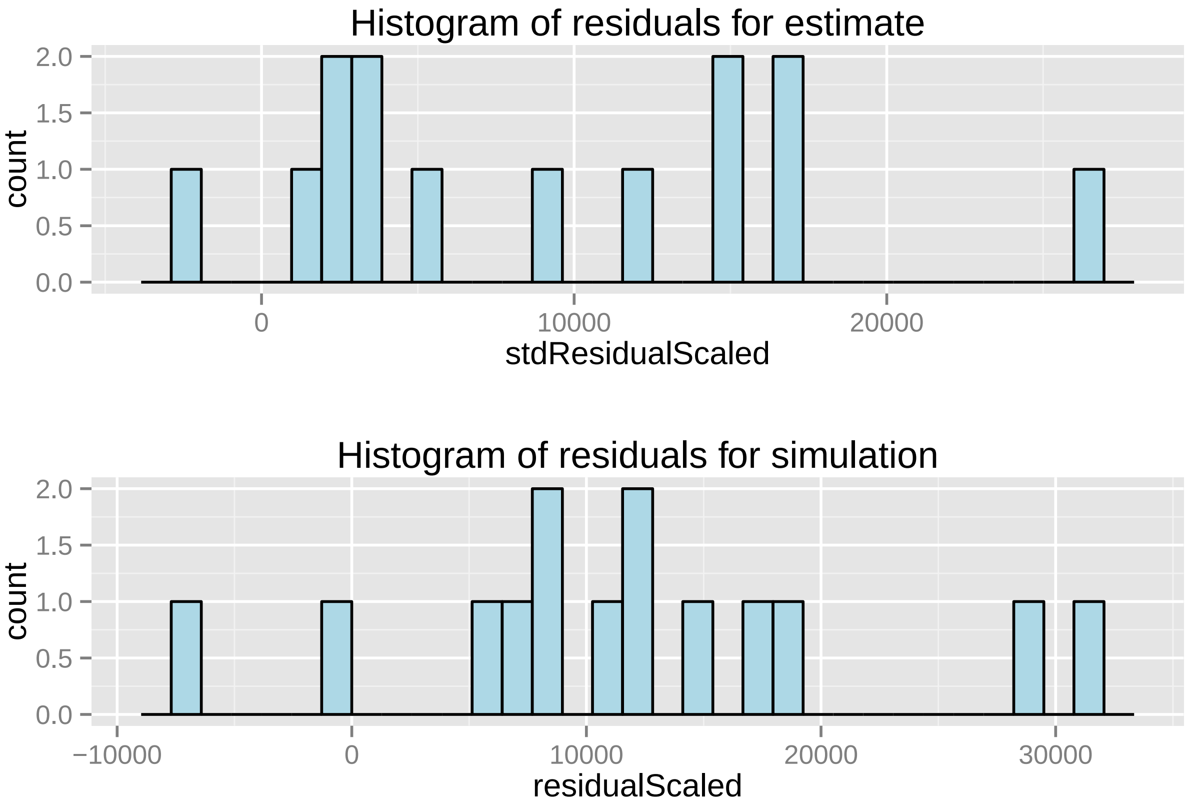




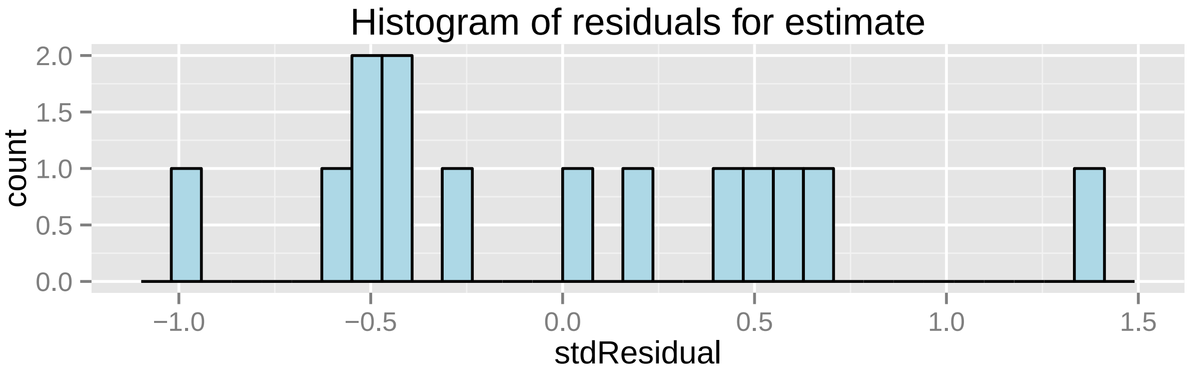
The simulation generates 1000 draws of the beta distribution and then selects the prediction based on the simulation with the lowest MSE, whereas the prediction performs a single prediction against the original data.

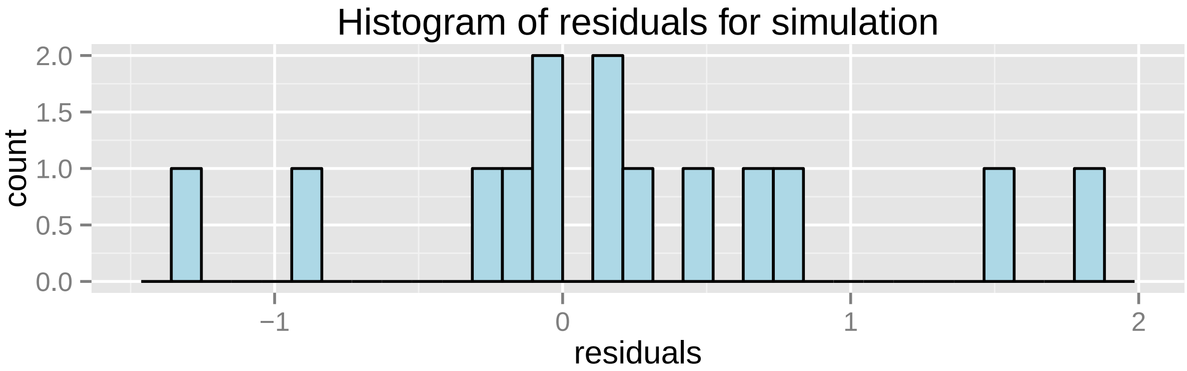
The residual of each approach is shown below.

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```





```
## [1] "MSE for prediction and simulation, both unscaled and scaled"
```

```
##        mse1 scaledmse1     simmse2 simscaledmse2
## 1 0.3941734   59351014 0.004243293      638916.1
```

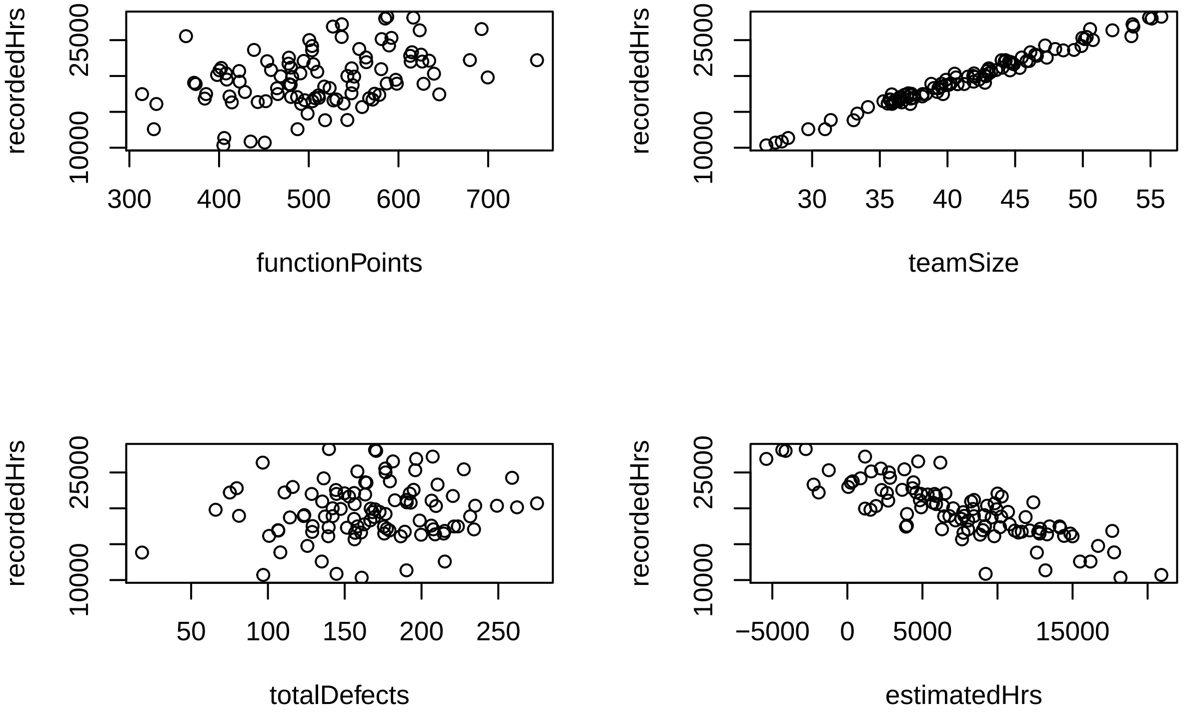
Based on the MSE alone, it suggests that it is possible to gain a more accurate estimate through the use of simulation, although the distribution of the simulation may be quite varied and would require a selection criteria for evaluating each simulation. It may be possible using the confidence bounds of the distribution to find a simulation that matches closely against the mean for the expected distribution in calculating an error, instead of the target variable directly.

Updating the Model.

In the case where we have additional observations it is possible to update the model. Unfortunately in the case of the software estimation domain we do not have available to us additional data, as further data collection is required, however in order to demonstrate the procedure, it is possible to use the model in a generative manner to create both the new dataset and the new target variable. This generated data set is drawn from the parameter space in the same manner as described above.

```
len <- 100
d2 <- test$betaConfData(test, test$sSqr,len, names= c("B1", "B2", "B3", "B4", "prob"))

newData = data.frame(b1=d2[,1]*test$xscale[1] + test$xcenter[1],
                     b2=d2[,2]*test$xscale[2] + test$xcenter[2],
                     b3=d2[,3]*test$xscale[3] + test$xcenter[3],
                     b4=d2[,4]*test$xscale[4] + test$xcenter[4])
colnames(newData) <- colnames(data)[1:4]
newY <- test$predictY(test, newData)
newY <- newY*test$yscale + test$ycenter
newData <- cbind(newData, data.frame(recordedHrs=newY))
```

Interestingly the generated data for the target variable appears to have a negative relationship with the estimated Hrs attribute, however appears to demonstrate a strong linear relationship with the teamSize, whilst only a slight linear relation with functionPoints, and even less of a relation with totalDefects.

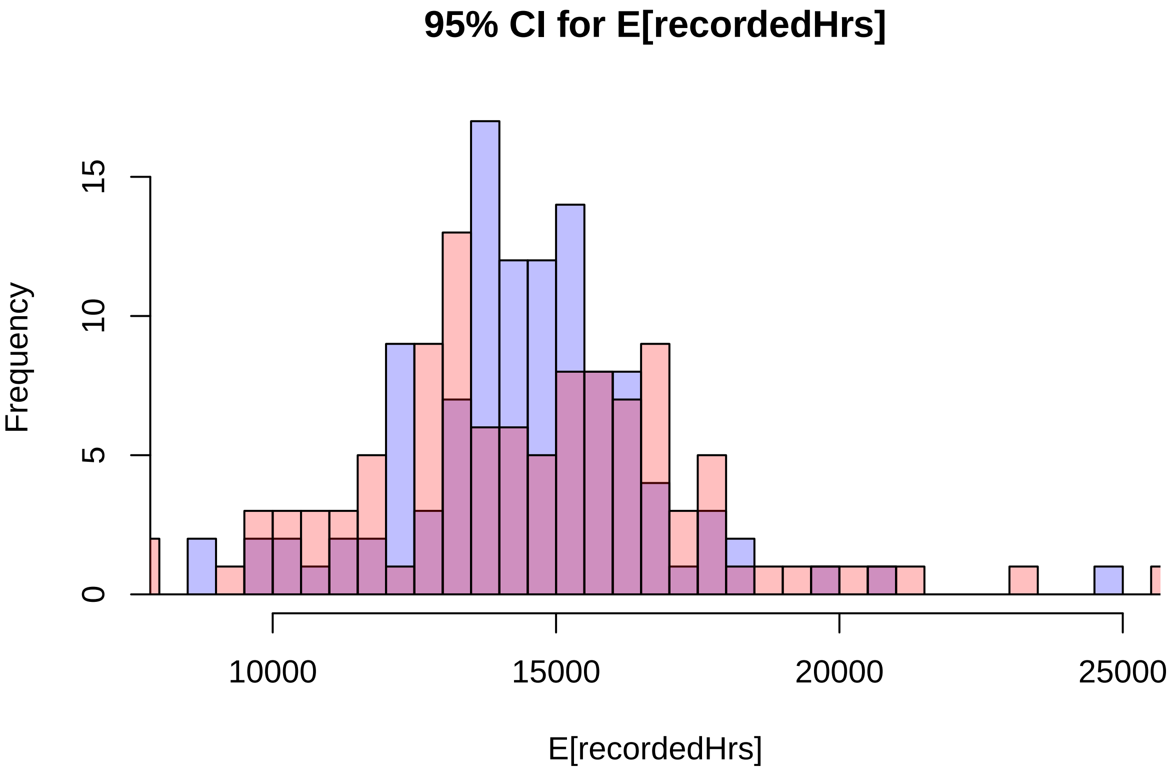


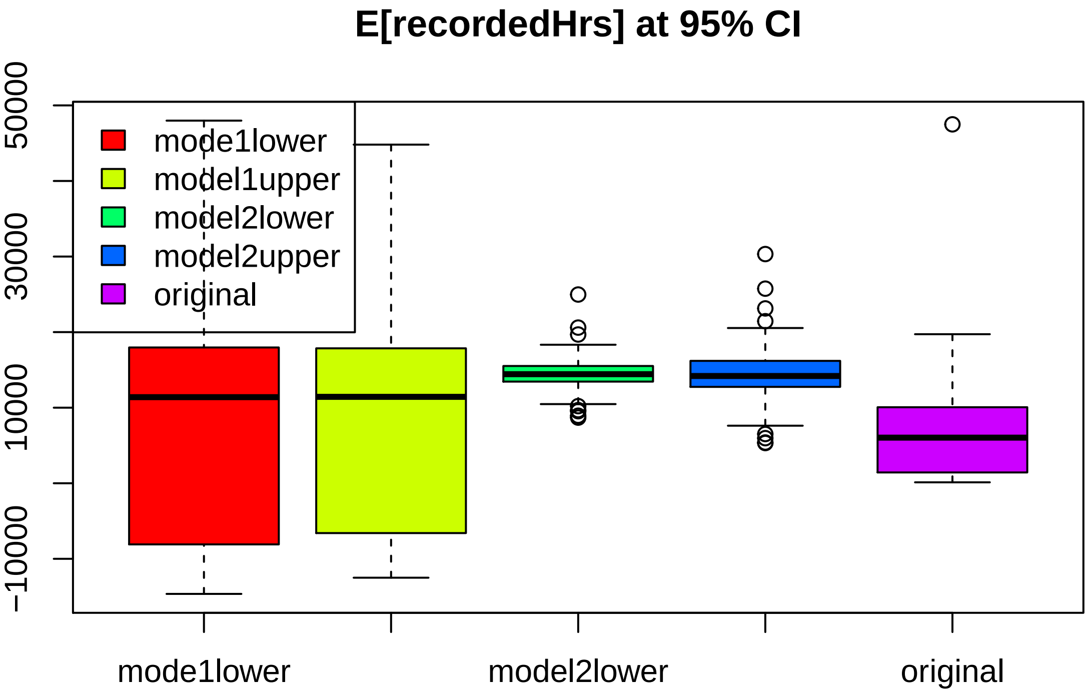
```
##  functionPoints     teamSize      totalDefects     estimatedHrs
##  Min.   :314.1   Min.   :26.62   Min.   : 18.03   Min.   :-5398
##  1st Qu.:453.2   1st Qu.:36.97   1st Qu.:139.49   1st Qu.: 3950
##  Median :508.2   Median :41.37   Median :166.81   Median : 7726
##  Mean   :512.4   Mean   :41.44   Mean   :165.85   Mean   : 7461
##  3rd Qu.:574.7   3rd Qu.:45.01   3rd Qu.:195.12   3rd Qu.:10895
##  Max.   :754.5   Max.   :55.80   Max.   :275.22   Max.   :20912
##   recordedHrs
##  Min.   :10330
##  1st Qu.:17035
##  Median :19483
##  Mean   :19674
##  3rd Qu.:22105
##  Max.   :28259
```

In order to update the model the new data is applied as follows.

```r
Xnew <- as.matrix(newData[,1:4])
Ynew <- as.matrix(data.frame(Y=newData[,5]))
updateModel <- test$applyUpdateRules(test, Ynew, Xnew)
```

Once the model has been updated we can also examine the confidence interval for the target variable





```
## [1] "Summary of original recordedHrs"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   121.8  1689.0  6032.0  9171.0  9923.0 47490.0
```

```
## [1] "Summary of original model CI recordedHrs"
```

```
##        mu             upper             lower
##  Min.   :-6175   Min.   :-12498   Min.   :-14641
##  1st Qu.:-1938   1st Qu.: -5952   1st Qu.: -7335
##  Median :11006   Median : 11433   Median : 11392
##  Mean   : 9171   Mean   :  9171   Mean   :  9171
##  3rd Qu.:16583   3rd Qu.: 17192   3rd Qu.: 17518
##  Max.   :35452   Max.   : 44811   Max.   : 47983
```

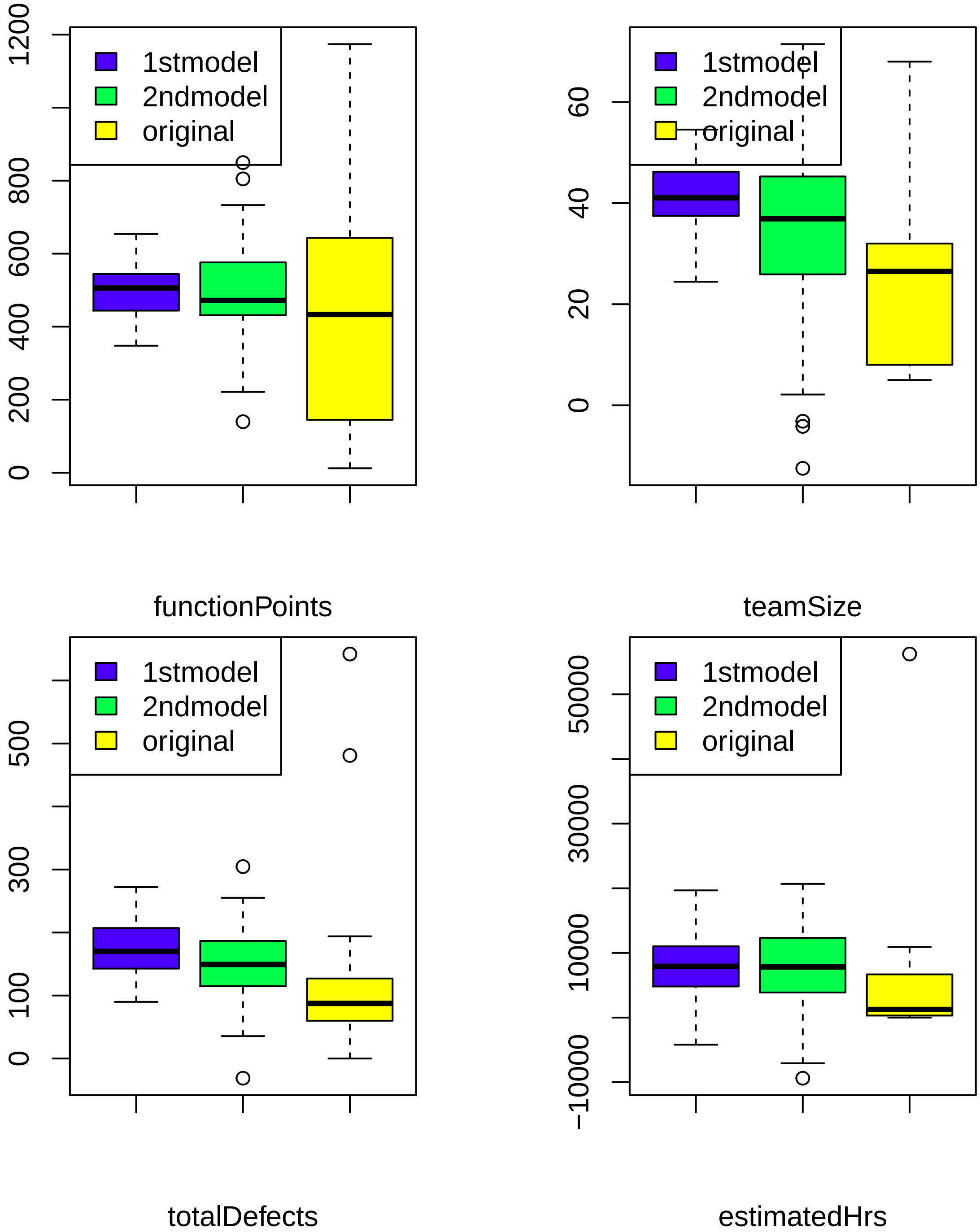
```
## [1] "Summary of updated model CI recordedHrs"
```

```
##      lower           upper
##  Min.   : 8685   Min.   : 5320
##  1st Qu.:13456   1st Qu.:12755
##  Median :14431   Median :14191
##  Mean   :14422   Mean   :14422
##  3rd Qu.:15510   3rd Qu.:16193
##  Max.   :24972   Max.   :30326
```

The distribution of the expected target variable has narrowed somewhat, this effect may be expected due to the use of data that was generated from the previous model.

Similarly the effect can be observed on the expected distributions of the predictors.

```
len <- 100
d2 <- updateModel$betaConfData(updateModel, updateModel$sSqr,len, names= c("B1", "B2", "B3", "B4", "prol

rConf2 = data.frame(b1=d2[,1]*updateModel$xscale[1] + updateModel$xcenter[1],
                   b2=d2[,2]*updateModel$xscale[2] + updateModel$xcenter[2],
                   b3=d2[,3]*updateModel$xscale[3] + updateModel$xcenter[3],
                   b4=d2[,4]*updateModel$xscale[4] + updateModel$xcenter[4])
colnames(rConf2) <- colnames(data)[1:4]
```

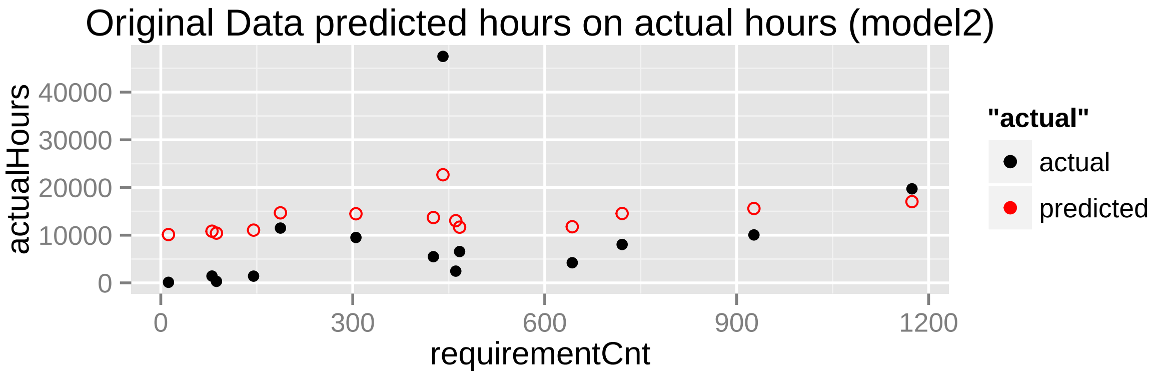


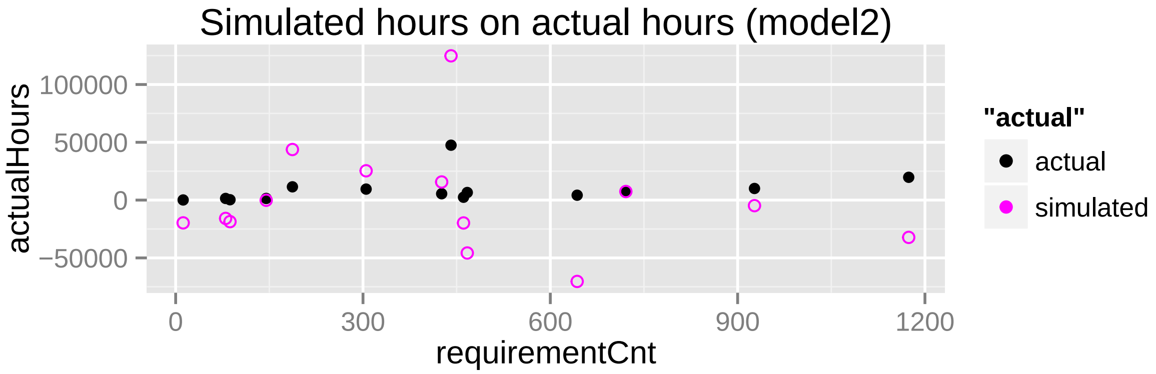
Interestingly there is some broadening of the predictor distribution visible as a result of the update.

In comparing the two models it is possible to evaluate predictions against the original data.

```
len <- 1000
X1 <- data[,xcols]
Y1 <- data[,ycol]

functionPoints <- data$functionPoints

## investigate the prediction
```

```
Ypred2 <- updateModel$predictY(updateModel, X1)
Ypred3 <- updateModel$yPredictivePosteriorSimulate(updateModel,X1, len)
```





In this case, it appears that the updated model may not be quite as accurate as the original model, and the MSE is comparatively larger for the predicted set (mse1, and scaledmse1). However in comparison of the mse generated on the simulated data these are equivalent.

```
## [1] "MSE for prediction and simulation, both unscaled and scaled"
```

```
##              mse1 scaledmse1     simmse2 simscaledmse2
## model1 0.3941734   59351014 0.004243293      638916.1
## model2 0.4941945   98386267 0.156448716    20146672.8
```

Model Evaluation

After having trained two models the question arises, how do we know whether Model A is better than Model B or vice versa. The Mean Square Error is a point estimate method of measuring the accuracy of the model, however additional approaches are available. One such approach is the **log pointwise predictive density** which is a measure of the sample predictive density that is proportional to the mean square error when the model is normal with con- stant variance [3]. This measure is taken as a measure of information gain given the model. The model with the higher lppd is taken to have better predictive generalisation.

$$lppd = log(E[p(y_i|\theta)])$$

where $\theta_1, ..., \theta_S$ are the model parameters and the distribution $p(y|\theta)$ is the joint posterior predictive distribution.

A second property is calculated to adjust for any overfitting that may be present in the model, this is known as the **widely applicable information criterion** (WAIC). This is calculated as

$$WAIC = 2\sum_{i=1}^{N}(\log(E[p(y_i|\theta)]) - E[\log p(y_i|\theta)])$$

the **expected log pointwise predictive density** is then calculated as the difference between the LPPD and WAIC.

In the example below, the data set for the original and the new data is combined, and a random sample of 50 examples is taken to calculate the ELPPD and the WAIC measures.

```
X2 <- rbind(data[,1:4], newData[,1:4])
Y2 <- rbind(data.frame(y=data[,5]), data.frame(y=newData[,5]))
indexes <- sample(nrow(X2), 50)
waic <- updateModel$WAIC(test, updateModel, Y2[indexes,], X2[indexes,])
```

```
## [1] "Comparison on original and new data"
```

```
##         modelEppd modelAdjEppd  modelWaic
## model1 0.02491659   -0.7371997 0.02491659
## model2 0.05754263   -0.4251392 0.05754263
```

Based on this assessment, the second model seems to indicate that it should be the better model for use generalisation over the data set. However this results due to the larger population of data from the second data set having being generated artificially.

References

1. Ekrem Kocaguneli , Gregory Gay , Tim Menzies , Ye Yang , Jacky W. Keung, When to use data from other projects for effort estimation, Proceedings of the IEEE/ACM international conference on Automated software engineering, September 20-24, 2010, Antwerp, Belgium, (also at https://ts.data61.csiro.au/publications/nictaabstracts/3723.pdf).

2. Box George E P,Tiao George C, Bayesian Inference in Statistical Analysis. Wiley 1992.

3. Gelman Andrew, Carlin John B, Stern Hal S, Dunson David B, Vehtari Aki, Rubin Donald B, Bayesian Data Analysis. Chapman and Hall/CRC Press Taylor and Francis Group 3rd edition, 2013