STAT2003 Analytics for Experimental and Simulated Data

**ASSIGNMENT 2**

5 pm, 6th November 2018

Due Date: 5 pm, 6^{th} Nov 2018

library(lhs)  
library(DiceDesign)  
library(FrF2)

## Loading required package: DoE.base

## Loading required package: grid

## Loading required package: conf.design

##   
## Attaching package: 'DoE.base'

## The following objects are masked from 'package:stats':  
##   
## aov, lm

## The following object is masked from 'package:graphics':  
##   
## plot.design

## The following object is masked from 'package:base':  
##   
## lengths

library(recoder)

## Loading required package: stringr

library(leaps)  
library(analogue)

## Loading required package: vegan

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.5-3

## analogue version 0.17-1

##   
## Attaching package: 'analogue'

## The following object is masked from 'package:conf.design':  
##   
## join

library(rriskDistributions)  
require(MASS)

## Loading required package: MASS

require(AlgDesign)

## Loading required package: AlgDesign

require(SPOT)

## Loading required package: SPOT

# Introduction

Your task is to develop a meta model (proxy model) for the water flow rate (WFR) through a borehole . The WFR is determined by eight input variables as described below

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input Variable | Variable Description | Lower Limit | Upper Limit | Code |
|  | Radius of borehole (m) | 0.05 | 0.15 |  |
|  | Radius of influence (m) | 100 | 50000 |  |
|  | Transitivity of upper aquifer () | 63070 | 115600 |  |
|  | Potentiometric head of upper aquifer () | 900 | 1110 |  |
|  | Transmissivity of lower aquifer () | 63.1 | 116 |  |
|  | Potentiometric head of lower aquifer (m) | 700 | 820 |  |
|  | Length of borehole (m) | 1120 | 1680 |  |
|  | Hydraulic conductivity of borehole () | 9855 | 12045 |  |

For any given settings of input variables, value of the water flow rate can be computed by function *WaterFlowRateInBorehole* given below

WaterFlowRateInBorehole<-function(x){  
(2\*(22/7)\*x[3]\*(x[4]-x[6]))/((log(x[2]/x[1]))\*(1+((2\*x[7]\*x[3])/((log(x[2]/x[1]))\*(x[1]^2)\*x[8]))+(x[3]/x[5]))) + rnorm(1,10,20)  
}  
  
#xmid=c(0.1,25000,89335,5055,89.5,760,1400,10950)  
set.seed(13)  
pbBore <- pb(12, 8, randomize = FALSE, seed = 13) #Creates Plackett Burmann 12 runs  
#Converts factors into numeric values for each column  
A <- as.numeric(pbBore$A)   
B <- as.numeric(pbBore$B)  
C <- as.numeric(pbBore$C)  
D <- as.numeric(pbBore$D)  
E <- as.numeric(pbBore$E)  
F1 <- as.numeric(pbBore$F)  
G <- as.numeric(pbBore$G)  
H <- as.numeric(pbBore$H)  
  
#According to whether the value is 2 or 1, convert it into the upper and lower limit accordingly  
rBore <- recoder(A, '== 2: 0.15; == 1: 0.05')  
rInf <- recoder(B, '== 2: 50000; == 1: 100')  
trans <- recoder(C, '== 2: 115600; == 1: 63070')  
potent <- recoder(D, '== 2: 1100; == 1: 900')  
transmis <- recoder(E, '== 2: 116; == 1: 63.1')  
potent2 <- recoder(F1, '== 2: 820; == 1: 700')  
lenBore <- recoder(G, '== 2: 1680; == 1: 1120')  
hydra <- recoder(H, '== 2: 12045; == 1: 9855')  
  
  
mtxPB <- matrix(c(rBore,rInf,trans, potent, transmis, potent2, lenBore, hydra), nrow = 12, ncol = 8) #Adds all results (upper and lower limit) into a matrix  
outMtx <- matrix(nrow = 12, ncol = 1) #Initiates a matrix to store results  
mtxPB

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]  
## [1,] 0.15 50000 63070 1100 116.0 820 1120 9855  
## [2,] 0.05 50000 115600 900 116.0 820 1680 9855  
## [3,] 0.15 100 115600 1100 63.1 820 1680 12045  
## [4,] 0.05 50000 63070 1100 116.0 700 1680 12045  
## [5,] 0.05 100 115600 900 116.0 820 1120 12045  
## [6,] 0.05 100 63070 1100 63.1 820 1680 9855  
## [7,] 0.15 100 63070 900 116.0 700 1680 12045  
## [8,] 0.15 50000 63070 900 63.1 820 1120 12045  
## [9,] 0.15 50000 115600 900 63.1 700 1680 9855  
## [10,] 0.05 50000 115600 1100 63.1 700 1120 12045  
## [11,] 0.15 100 115600 1100 116.0 700 1120 9855  
## [12,] 0.05 100 63070 900 63.1 700 1120 9855

#Generates results of upper and lower limit matrix and adds it to the results matrix  
for (i in seq(1,12)){  
 outMtx[i,1] <- WaterFlowRateInBorehole(mtxPB[i,])  
}  
outMtx

## [,1]  
## [1,] 193.435074  
## [2,] 8.078586  
## [3,] 186.291376  
## [4,] 36.255496  
## [5,] 39.604517  
## [6,] 31.204480  
## [7,] 135.532142  
## [8,] 74.123560  
## [9,] 84.565843  
## [10,] 65.803385  
## [11,] 235.642365  
## [12,] 33.046273

#WaterFlowRateInBorehole(xmid)

reformatDF <- data.frame(mtxPB)  
reformatDF <- cbind(reformatDF,outMtx)  
reformatDF #matrix containing results and min/max ranges

## X1 X2 X3 X4 X5 X6 X7 X8 outMtx  
## 1 0.15 50000 63070 1100 116.0 820 1120 9855 193.435074  
## 2 0.05 50000 115600 900 116.0 820 1680 9855 8.078586  
## 3 0.15 100 115600 1100 63.1 820 1680 12045 186.291376  
## 4 0.05 50000 63070 1100 116.0 700 1680 12045 36.255496  
## 5 0.05 100 115600 900 116.0 820 1120 12045 39.604517  
## 6 0.05 100 63070 1100 63.1 820 1680 9855 31.204480  
## 7 0.15 100 63070 900 116.0 700 1680 12045 135.532142  
## 8 0.15 50000 63070 900 63.1 820 1120 12045 74.123560  
## 9 0.15 50000 115600 900 63.1 700 1680 9855 84.565843  
## 10 0.05 50000 115600 1100 63.1 700 1120 12045 65.803385  
## 11 0.15 100 115600 1100 116.0 700 1120 9855 235.642365  
## 12 0.05 100 63070 900 63.1 700 1120 9855 33.046273

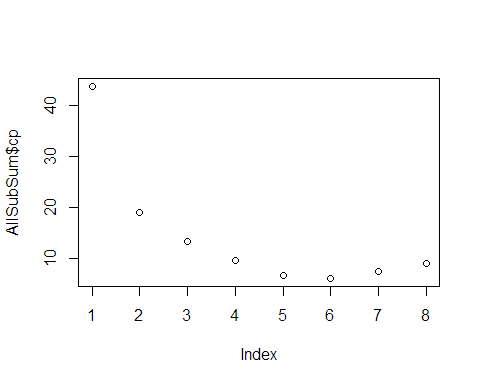
AllSubsets <- regsubsets(outMtx~., data = reformatDF, nvmax = 11, nbest = 1)  
AllSubSum <- summary(AllSubsets)  
AllSubSum$outmat

## X1 X2 X3 X4 X5 X6 X7 X8   
## 1 ( 1 ) "\*" " " " " " " " " " " " " " "  
## 2 ( 1 ) "\*" " " " " "\*" " " " " " " " "  
## 3 ( 1 ) "\*" "\*" " " "\*" " " " " " " " "  
## 4 ( 1 ) "\*" "\*" " " "\*" "\*" " " " " " "  
## 5 ( 1 ) "\*" "\*" " " "\*" "\*" " " "\*" " "  
## 6 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " "\*" " "  
## 7 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" " "  
## 8 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"

summary(AllSubSum)

## Length Class Mode   
## which 72 -none- logical   
## rsq 8 -none- numeric   
## rss 8 -none- numeric   
## adjr2 8 -none- numeric   
## cp 8 -none- numeric   
## bic 8 -none- numeric   
## outmat 64 -none- character  
## obj 28 regsubsets list

plot(AllSubSum$cp)



boreAOV <- aov(outMtx~., data = reformatDF)  
anova(boreAOV)

## Analysis of Variance Table  
##   
## Response: outMtx  
## Df Sum Sq Mean Sq F value Pr(>F)   
## X1 1 40321 40321 92.7320 0.002377 \*\*  
## X2 1 3302 3302 7.5941 0.070403 .   
## X3 1 1129 1129 2.5962 0.205508   
## X4 1 11636 11636 26.7618 0.014017 \*   
## X5 1 2509 2509 5.7700 0.095694 .   
## X6 1 281 281 0.6471 0.479988   
## X7 1 2126 2126 4.8896 0.113970   
## X8 1 195 195 0.4483 0.551086   
## Residuals 3 1304 435   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AllSubSum$outmat

## X1 X2 X3 X4 X5 X6 X7 X8   
## 1 ( 1 ) "\*" " " " " " " " " " " " " " "  
## 2 ( 1 ) "\*" " " " " "\*" " " " " " " " "  
## 3 ( 1 ) "\*" "\*" " " "\*" " " " " " " " "  
## 4 ( 1 ) "\*" "\*" " " "\*" "\*" " " " " " "  
## 5 ( 1 ) "\*" "\*" " " "\*" "\*" " " "\*" " "  
## 6 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " "\*" " "  
## 7 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" " "  
## 8 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"

# creating our model using stewpise selection  
SimplestModel <- lm(outMtx ~ 1, data = reformatDF)  
FullModel <- lm(outMtx ~., data = reformatDF)  
Forward <- step(SimplestModel, scope = formula(FullModel), direction = 'forward')

## Start: AIC=104.75  
## outMtx ~ 1  
##   
## Df Sum of Sq RSS AIC  
## + X1 1 40321 22483 94.427  
## + X4 1 11636 51168 104.296  
## <none> 62804 104.754  
## + X2 1 3302 59502 106.106  
## + X5 1 2509 60296 106.265  
## + X7 1 2126 60678 106.341  
## + X3 1 1129 61676 106.537  
## + X6 1 281 62523 106.701  
## + X8 1 195 62610 106.717  
##   
## Step: AIC=94.43  
## outMtx ~ X1  
##   
## Df Sum of Sq RSS AIC  
## + X4 1 11636.5 10847 87.680  
## <none> 22483 94.427  
## + X2 1 3302.0 19181 94.521  
## + X5 1 2508.9 19974 95.007  
## + X7 1 2126.1 20357 95.235  
## + X3 1 1128.9 21354 95.809  
## + X6 1 281.4 22202 96.276  
## + X8 1 194.9 22288 96.323  
##   
## Step: AIC=87.68  
## outMtx ~ X1 + X4  
##   
## Df Sum of Sq RSS AIC  
## + X2 1 3302.0 7544.6 85.324  
## + X5 1 2508.9 8337.7 86.524  
## + X7 1 2126.1 8720.6 87.062  
## <none> 10846.6 87.680  
## + X3 1 1128.9 9717.8 88.362  
## + X6 1 281.4 10565.2 89.365  
## + X8 1 194.9 10651.7 89.463  
##   
## Step: AIC=85.32  
## outMtx ~ X1 + X4 + X2  
##   
## Df Sum of Sq RSS AIC  
## + X5 1 2508.90 5035.7 82.473  
## + X7 1 2126.07 5418.5 83.352  
## <none> 7544.6 85.324  
## + X3 1 1128.87 6415.7 85.379  
## + X6 1 281.38 7263.2 86.868  
## + X8 1 194.91 7349.7 87.010  
##   
## Step: AIC=82.47  
## outMtx ~ X1 + X4 + X2 + X5  
##   
## Df Sum of Sq RSS AIC  
## + X7 1 2126.07 2909.6 77.890  
## + X3 1 1128.87 3906.8 81.427  
## <none> 5035.7 82.473  
## + X6 1 281.38 4754.3 83.783  
## + X8 1 194.91 4840.8 83.999  
##   
## Step: AIC=77.89  
## outMtx ~ X1 + X4 + X2 + X5 + X7  
##   
## Df Sum of Sq RSS AIC  
## + X3 1 1128.87 1780.7 73.998  
## <none> 2909.6 77.890  
## + X6 1 281.38 2628.2 78.670  
## + X8 1 194.91 2714.7 79.058  
##   
## Step: AIC=74  
## outMtx ~ X1 + X4 + X2 + X5 + X7 + X3  
##   
## Df Sum of Sq RSS AIC  
## + X6 1 281.38 1499.4 73.935  
## <none> 1780.7 73.998  
## + X8 1 194.91 1585.8 74.607  
##   
## Step: AIC=73.93  
## outMtx ~ X1 + X4 + X2 + X5 + X7 + X3 + X6  
##   
## Df Sum of Sq RSS AIC  
## <none> 1499.4 73.935  
## + X8 1 194.91 1304.5 74.264

summary(Forward)

##   
## Call:  
## lm.default(formula = outMtx ~ X1 + X4 + X2 + X5 + X7 + X3 + X6,   
## data = reformatDF)  
##   
## Residuals:  
## 1 2 3 4 5 6 7 8 9   
## 14.056 14.136 9.878 -10.254 -14.136 -9.878 2.194 -14.056 -6.076   
## 10 11 12   
## 2.194 -5.996 17.938   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.711e+02 9.911e+01 -2.735 0.052158 .   
## X1 1.159e+03 1.118e+02 10.372 0.000488 \*\*\*  
## X4 3.114e-01 5.589e-02 5.572 0.005085 \*\*   
## X2 -6.649e-04 2.240e-04 -2.968 0.041224 \*   
## X5 5.467e-01 2.113e-01 2.587 0.060867 .   
## X7 -4.754e-02 1.996e-02 -2.382 0.075858 .   
## X3 3.693e-04 2.128e-04 1.735 0.157684   
## X6 -8.071e-02 9.315e-02 -0.866 0.435144   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 19.36 on 4 degrees of freedom  
## Multiple R-squared: 0.9761, Adjusted R-squared: 0.9343   
## F-statistic: 23.36 on 7 and 4 DF, p-value: 0.004312

print("Forward")

## [1] "Forward"

coef(Forward)

## (Intercept) X1 X4 X2 X5   
## -2.711020e+02 1.159329e+03 3.114010e-01 -6.648604e-04 5.466706e-01   
## X7 X3 X6   
## -4.753787e-02 3.692780e-04 -8.070543e-02

Backward <- step(FullModel, direction = 'backward')

## Start: AIC=74.26  
## outMtx ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8  
##   
## Df Sum of Sq RSS AIC  
## - X8 1 195 1499 73.935  
## <none> 1304 74.264  
## - X6 1 281 1586 74.607  
## - X3 1 1129 2433 79.745  
## - X7 1 2126 3431 83.867  
## - X5 1 2509 3813 85.136  
## - X2 1 3302 4606 87.404  
## - X4 1 11636 12941 99.799  
## - X1 1 40321 41626 113.819  
##   
## Step: AIC=73.93  
## outMtx ~ X1 + X2 + X3 + X4 + X5 + X6 + X7  
##   
## Df Sum of Sq RSS AIC  
## <none> 1499 73.935  
## - X6 1 281 1781 73.998  
## - X3 1 1129 2628 78.670  
## - X7 1 2126 3625 82.530  
## - X5 1 2509 4008 83.734  
## - X2 1 3302 4801 85.901  
## - X4 1 11636 13136 97.978  
## - X1 1 40321 41821 111.875

print("Backward")

## [1] "Backward"

coef(Backward)

## (Intercept) X1 X2 X3 X4   
## -2.711020e+02 1.159329e+03 -6.648604e-04 3.692780e-04 3.114010e-01   
## X5 X6 X7   
## 5.466706e-01 -8.070543e-02 -4.753787e-02

Step <- step(SimplestModel, scope = formula(FullModel))

## Start: AIC=104.75  
## outMtx ~ 1  
##   
## Df Sum of Sq RSS AIC  
## + X1 1 40321 22483 94.427  
## + X4 1 11636 51168 104.296  
## <none> 62804 104.754  
## + X2 1 3302 59502 106.106  
## + X5 1 2509 60296 106.265  
## + X7 1 2126 60678 106.341  
## + X3 1 1129 61676 106.537  
## + X6 1 281 62523 106.701  
## + X8 1 195 62610 106.717  
##   
## Step: AIC=94.43  
## outMtx ~ X1  
##   
## Df Sum of Sq RSS AIC  
## + X4 1 11636 10847 87.680  
## <none> 22483 94.427  
## + X2 1 3302 19181 94.521  
## + X5 1 2509 19974 95.007  
## + X7 1 2126 20357 95.235  
## + X3 1 1129 21354 95.809  
## + X6 1 281 22202 96.276  
## + X8 1 195 22288 96.323  
## - X1 1 40321 62804 104.754  
##   
## Step: AIC=87.68  
## outMtx ~ X1 + X4  
##   
## Df Sum of Sq RSS AIC  
## + X2 1 3302 7545 85.324  
## + X5 1 2509 8338 86.524  
## + X7 1 2126 8721 87.062  
## <none> 10847 87.680  
## + X3 1 1129 9718 88.362  
## + X6 1 281 10565 89.365  
## + X8 1 195 10652 89.463  
## - X4 1 11636 22483 94.427  
## - X1 1 40321 51168 104.296  
##   
## Step: AIC=85.32  
## outMtx ~ X1 + X4 + X2  
##   
## Df Sum of Sq RSS AIC  
## + X5 1 2509 5036 82.473  
## + X7 1 2126 5419 83.352  
## <none> 7545 85.324  
## + X3 1 1129 6416 85.379  
## + X6 1 281 7263 86.868  
## + X8 1 195 7350 87.010  
## - X2 1 3302 10847 87.680  
## - X4 1 11636 19181 94.521  
## - X1 1 40321 47866 105.495  
##   
## Step: AIC=82.47  
## outMtx ~ X1 + X4 + X2 + X5  
##   
## Df Sum of Sq RSS AIC  
## + X7 1 2126 2910 77.890  
## + X3 1 1129 3907 81.427  
## <none> 5036 82.473  
## + X6 1 281 4754 83.783  
## + X8 1 195 4841 83.999  
## - X5 1 2509 7545 85.324  
## - X2 1 3302 8338 86.524  
## - X4 1 11636 16672 94.839  
## - X1 1 40321 45357 106.849  
##   
## Step: AIC=77.89  
## outMtx ~ X1 + X4 + X2 + X5 + X7  
##   
## Df Sum of Sq RSS AIC  
## + X3 1 1129 1781 73.998  
## <none> 2910 77.890  
## + X6 1 281 2628 78.670  
## + X8 1 195 2715 79.058  
## - X7 1 2126 5036 82.473  
## - X5 1 2509 5419 83.352  
## - X2 1 3302 6212 84.991  
## - X4 1 11636 14546 95.202  
## - X1 1 40321 43231 108.273  
##   
## Step: AIC=74  
## outMtx ~ X1 + X4 + X2 + X5 + X7 + X3  
##   
## Df Sum of Sq RSS AIC  
## + X6 1 281 1499 73.935  
## <none> 1781 73.998  
## + X8 1 195 1586 74.607  
## - X3 1 1129 2910 77.890  
## - X7 1 2126 3907 81.427  
## - X5 1 2509 4290 82.549  
## - X2 1 3302 5083 84.584  
## - X4 1 11636 13417 96.233  
## - X1 1 40321 42102 109.955  
##   
## Step: AIC=73.93  
## outMtx ~ X1 + X4 + X2 + X5 + X7 + X3 + X6  
##   
## Df Sum of Sq RSS AIC  
## <none> 1499 73.935  
## - X6 1 281 1781 73.998  
## + X8 1 195 1304 74.264  
## - X3 1 1129 2628 78.670  
## - X7 1 2126 3625 82.530  
## - X5 1 2509 4008 83.734  
## - X2 1 3302 4801 85.901  
## - X4 1 11636 13136 97.978  
## - X1 1 40321 41821 111.875

print("Step")

## [1] "Step"

coef(Step)

## (Intercept) X1 X4 X2 X5   
## -2.711020e+02 1.159329e+03 3.114010e-01 -6.648604e-04 5.466706e-01   
## X7 X3 X6   
## -4.753787e-02 3.692780e-04 -8.070543e-02

set.seed(13)  
xMatrix <- gen.factorial(levels = 3, nVars = 4, varNames = c("X1","X2","X4","X5"))  
  
xd <- optFederov(~X1+X2+X4+X5, data = xMatrix, 50)  
#Cm <- mean()  
X1\_m <- mean(range(0.05, 0.15))  
X2\_m <- mean(range(100, 50000))  
X3\_m <- mean(range(63070, 115600))  
X4\_m <- mean(range(900, 1110))  
X5\_m <- mean(range(63.1, 116))  
X6\_m <- mean(range(700, 820))  
X7\_m <- mean(range(1120, 1680))  
X8\_m <- mean(range(9855, 12045))  
  
X1 <- xd$design$X1  
X2 <- xd$design$X2  
X4 <- xd$design$X4  
X5 <- xd$design$X5  
  
X1 <- recoder(X1, '== 1: 0.15; == -1: 0.05; == 0: X1\_m')  
X2 <- recoder(X2, '== 1: 100; == -1: 50000; == 0: X2\_m')  
X4 <- recoder(X4, '== 1: 900; == -1: 1110; == 0: X4\_m')  
X5 <- recoder(X5, '== 1: 63.1; == -1: 116; == 0: X5\_m')  
  
num\_rows <- nrow(xd$design)  
  
X3 <- rep(X3\_m, num\_rows)  
X6 <- rep(X6\_m, num\_rows)  
X7 <- rep(X7\_m, num\_rows)  
X8 <- rep(X8\_m, num\_rows)  
  
mtxDopt <- matrix(c(X1,X2,X3,X4,X5,X6,X7,X8), ncol = 8)  
mtxDopt

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]  
## [1,] 0.05 50000 89335 1110 116.00 760 1400 10950  
## [2,] 0.10 50000 89335 1110 116.00 760 1400 10950  
## [3,] 0.15 50000 89335 1110 116.00 760 1400 10950  
## [4,] 0.05 25050 89335 1110 116.00 760 1400 10950  
## [5,] 0.15 25050 89335 1110 116.00 760 1400 10950  
## [6,] 0.05 100 89335 1110 116.00 760 1400 10950  
## [7,] 0.10 100 89335 1110 116.00 760 1400 10950  
## [8,] 0.15 100 89335 1110 116.00 760 1400 10950  
## [9,] 0.05 50000 89335 1005 116.00 760 1400 10950  
## [10,] 0.15 50000 89335 1005 116.00 760 1400 10950  
## [11,] 0.05 100 89335 1005 116.00 760 1400 10950  
## [12,] 0.15 100 89335 1005 116.00 760 1400 10950  
## [13,] 0.05 50000 89335 900 116.00 760 1400 10950  
## [14,] 0.10 50000 89335 900 116.00 760 1400 10950  
## [15,] 0.15 50000 89335 900 116.00 760 1400 10950  
## [16,] 0.05 25050 89335 900 116.00 760 1400 10950  
## [17,] 0.15 25050 89335 900 116.00 760 1400 10950  
## [18,] 0.05 100 89335 900 116.00 760 1400 10950  
## [19,] 0.10 100 89335 900 116.00 760 1400 10950  
## [20,] 0.15 100 89335 900 116.00 760 1400 10950  
## [21,] 0.05 50000 89335 1110 89.55 760 1400 10950  
## [22,] 0.15 50000 89335 1110 89.55 760 1400 10950  
## [23,] 0.05 25050 89335 1110 89.55 760 1400 10950  
## [24,] 0.05 100 89335 1110 89.55 760 1400 10950  
## [25,] 0.15 100 89335 1110 89.55 760 1400 10950  
## [26,] 0.15 50000 89335 1005 89.55 760 1400 10950  
## [27,] 0.05 50000 89335 900 89.55 760 1400 10950  
## [28,] 0.15 50000 89335 900 89.55 760 1400 10950  
## [29,] 0.05 100 89335 900 89.55 760 1400 10950  
## [30,] 0.15 100 89335 900 89.55 760 1400 10950  
## [31,] 0.05 50000 89335 1110 63.10 760 1400 10950  
## [32,] 0.10 50000 89335 1110 63.10 760 1400 10950  
## [33,] 0.15 50000 89335 1110 63.10 760 1400 10950  
## [34,] 0.05 25050 89335 1110 63.10 760 1400 10950  
## [35,] 0.15 25050 89335 1110 63.10 760 1400 10950  
## [36,] 0.05 100 89335 1110 63.10 760 1400 10950  
## [37,] 0.10 100 89335 1110 63.10 760 1400 10950  
## [38,] 0.15 100 89335 1110 63.10 760 1400 10950  
## [39,] 0.05 50000 89335 1005 63.10 760 1400 10950  
## [40,] 0.15 50000 89335 1005 63.10 760 1400 10950  
## [41,] 0.05 100 89335 1005 63.10 760 1400 10950  
## [42,] 0.15 100 89335 1005 63.10 760 1400 10950  
## [43,] 0.05 50000 89335 900 63.10 760 1400 10950  
## [44,] 0.10 50000 89335 900 63.10 760 1400 10950  
## [45,] 0.15 50000 89335 900 63.10 760 1400 10950  
## [46,] 0.05 25050 89335 900 63.10 760 1400 10950  
## [47,] 0.15 25050 89335 900 63.10 760 1400 10950  
## [48,] 0.05 100 89335 900 63.10 760 1400 10950  
## [49,] 0.10 100 89335 900 63.10 760 1400 10950  
## [50,] 0.15 100 89335 900 63.10 760 1400 10950

test\_mtx <- data.frame(cbind(runif(50, 0.05, 0.15),  
 runif(50, 100, 50000),  
 runif(50, 63070, 115600),  
 runif(50, 900, 1110),  
 runif(50, 63.1, 116),  
 runif(50, 700, 820),  
 runif(50, 1120, 1680),  
 runif(50, 9855, 12045),   
 ncol = 8,   
 nrow = 50))

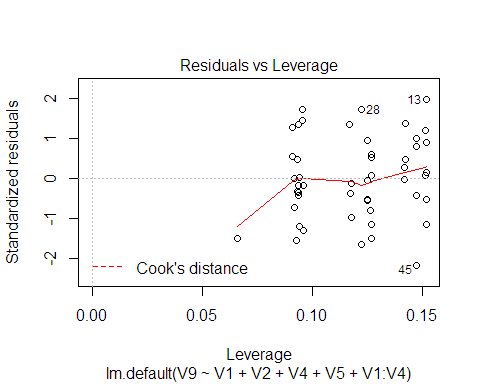
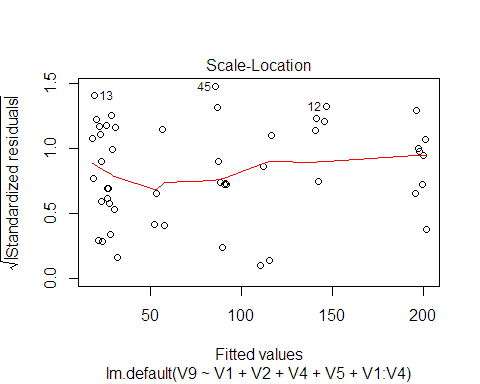
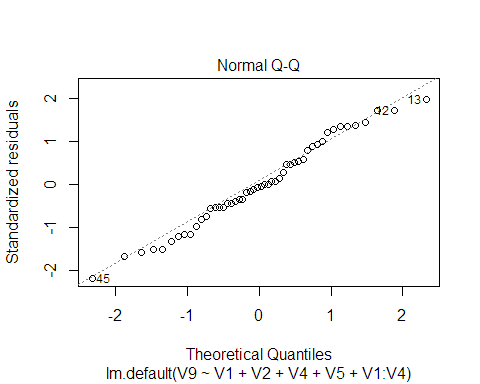
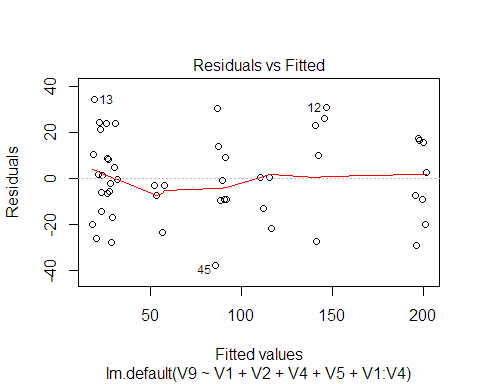
mtxRes <- matrix(nrow = num\_rows, ncol = 1)  
for (i in seq(1,50)){  
 mtxRes[i,1] <- WaterFlowRateInBorehole(mtxDopt[i,])  
}  
  
mtx\_opt <- cbind(mtxDopt, mtxRes)  
eg <- as.data.frame(mtx\_opt)  
eg

## V1 V2 V3 V4 V5 V6 V7 V8 V9  
## 1 0.05 50000 89335 1110 116.00 760 1400 10950 35.7188582  
## 2 0.10 50000 89335 1110 116.00 760 1400 10950 98.9716708  
## 3 0.15 50000 89335 1110 116.00 760 1400 10950 214.3965683  
## 4 0.05 25050 89335 1110 116.00 760 1400 10950 12.4251768  
## 5 0.15 25050 89335 1110 116.00 760 1400 10950 189.9934486  
## 6 0.05 100 89335 1110 116.00 760 1400 10950 31.4740592  
## 7 0.10 100 89335 1110 116.00 760 1400 10950 94.9539322  
## 8 0.15 100 89335 1110 116.00 760 1400 10950 203.9537535  
## 9 0.05 50000 89335 1005 116.00 760 1400 10950 17.2433049  
## 10 0.15 50000 89335 1005 116.00 760 1400 10950 152.1240618  
## 11 0.05 100 89335 1005 116.00 760 1400 10950 22.1262529  
## 12 0.15 100 89335 1005 116.00 760 1400 10950 177.8534347  
## 13 0.05 50000 89335 900 116.00 760 1400 10950 53.9852933  
## 14 0.10 50000 89335 900 116.00 760 1400 10950 45.9528744  
## 15 0.15 50000 89335 900 116.00 760 1400 10950 101.5419699  
## 16 0.05 25050 89335 900 116.00 760 1400 10950 23.3483637  
## 17 0.15 25050 89335 900 116.00 760 1400 10950 88.7634326  
## 18 0.05 100 89335 900 116.00 760 1400 10950 25.5448275  
## 19 0.10 100 89335 900 116.00 760 1400 10950 55.1065803  
## 20 0.15 100 89335 900 116.00 760 1400 10950 83.0052975  
## 21 0.05 50000 89335 1110 89.55 760 1400 10950 20.0381874  
## 22 0.15 50000 89335 1110 89.55 760 1400 10950 166.8118018  
## 23 0.05 25050 89335 1110 89.55 760 1400 10950 0.8248175  
## 24 0.05 100 89335 1110 89.55 760 1400 10950 55.2005578  
## 25 0.15 100 89335 1110 89.55 760 1400 10950 180.6091489  
## 26 0.15 50000 89335 1005 89.55 760 1400 10950 114.0590134  
## 27 0.05 50000 89335 900 89.55 760 1400 10950 29.3930545  
## 28 0.15 50000 89335 900 89.55 760 1400 10950 117.3684590  
## 29 0.05 100 89335 900 89.55 760 1400 10950 9.1445048  
## 30 0.15 100 89335 900 89.55 760 1400 10950 100.6254849  
## 31 0.05 50000 89335 1110 63.10 760 1400 10950 50.1785212  
## 32 0.10 50000 89335 1110 63.10 760 1400 10950 111.0059534  
## 33 0.15 50000 89335 1110 63.10 760 1400 10950 188.1818459  
## 34 0.05 25050 89335 1110 63.10 760 1400 10950 26.2885103  
## 35 0.15 25050 89335 1110 63.10 760 1400 10950 214.5743702  
## 36 0.05 100 89335 1110 63.10 760 1400 10950 35.5448230  
## 37 0.10 100 89335 1110 63.10 760 1400 10950 115.6871842  
## 38 0.15 100 89335 1110 63.10 760 1400 10950 215.6994909  
## 39 0.05 50000 89335 1005 63.10 760 1400 10950 46.6962566  
## 40 0.15 50000 89335 1005 63.10 760 1400 10950 163.9296322  
## 41 0.05 100 89335 1005 63.10 760 1400 10950 35.2488651  
## 42 0.15 100 89335 1005 63.10 760 1400 10950 171.3911478  
## 43 0.05 50000 89335 900 63.10 760 1400 10950 -1.9577065  
## 44 0.10 50000 89335 900 63.10 760 1400 10950 49.0585990  
## 45 0.15 50000 89335 900 63.10 760 1400 10950 48.2593530  
## 46 0.05 25050 89335 900 63.10 760 1400 10950 -5.8152463  
## 47 0.15 25050 89335 900 63.10 760 1400 10950 78.7319959  
## 48 0.05 100 89335 900 63.10 760 1400 10950 43.9089782  
## 49 0.10 100 89335 900 63.10 760 1400 10950 33.2239994  
## 50 0.15 100 89335 900 63.10 760 1400 10950 81.6217622

optLm <- lm(V9~V1+V2+V4+V5+V1:V4, data = eg)  
optPred <- predict(optLm)  
anova(optLm)

## Analysis of Variance Table  
##   
## Response: V9  
## Df Sum Sq Mean Sq F value Pr(>F)   
## V1 1 147258 147258 414.0120 < 2.2e-16 \*\*\*  
## V2 1 210 210 0.5894 0.4468   
## V4 1 33794 33794 95.0100 1.462e-12 \*\*\*  
## V5 1 18 18 0.0513 0.8218   
## V1:V4 1 21266 21266 59.7893 9.804e-10 \*\*\*  
## Residuals 44 15650 356   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

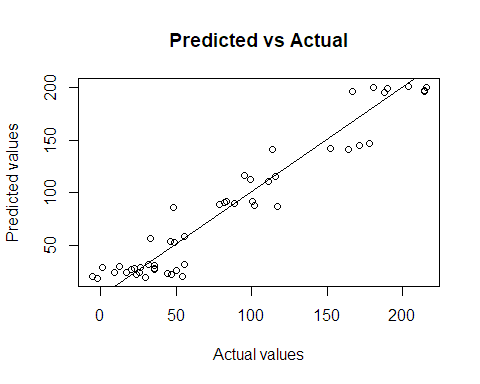
plot(optLm)



summary(optLm)

##   
## Call:  
## lm.default(formula = V9 ~ V1 + V2 + V4 + V5 + V1:V4, data = eg)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -37.880 -9.557 -0.725 13.157 34.392   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.715e+02 6.965e+01 2.463 0.01777 \*   
## V1 -3.677e+03 6.333e+02 -5.806 6.49e-07 \*\*\*  
## V2 -9.069e-05 1.181e-04 -0.768 0.44669   
## V4 -2.048e-01 6.793e-02 -3.015 0.00426 \*\*   
## V5 2.554e-02 1.127e-01 0.227 0.82181   
## V1:V4 4.840e+00 6.260e-01 7.732 9.80e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.86 on 44 degrees of freedom  
## Multiple R-squared: 0.9283, Adjusted R-squared: 0.9201   
## F-statistic: 113.9 on 5 and 44 DF, p-value: < 2.2e-16

newMtx <- as.matrix(cbind(mtxRes, optPred))  
  
RMSEP <- function(y){  
 sqrt((sum((y[,2]-y[,1])^2))/50)  
}  
  
RMSEP4 <- RMSEP(newMtx)  
  
plot(newMtx, ylab = "Predicted values", xlab = "Actual values", main = "Predicted vs Actual")  
abline(1,1)



set.seed(13)  
xMatrix <- gen.factorial(levels = 3, nVars = 6, varNames = c("X1","X2","X3","X4","X5","X7"))  
  
xd <- optFederov(~X1+X2+X3+X4+X5+X7, data = xMatrix, 50)  
#Cm <- mean()  
X1\_m <- mean(range(0.05, 0.15))  
X2\_m <- mean(range(100, 50000))  
X3\_m <- mean(range(63070, 115600))  
X4\_m <- mean(range(900, 1110))  
X5\_m <- mean(range(63.1, 116))  
X6\_m <- mean(range(700, 820))  
X7\_m <- mean(range(1120, 1680))  
X8\_m <- mean(range(9855, 12045))  
  
X1 <- xd$design$X1  
X2 <- xd$design$X2  
X3 <- xd$design$X3  
X4 <- xd$design$X4  
X5 <- xd$design$X5  
X7 <- xd$design$X7  
  
X1 <- recoder(X1, '== 1: 0.15; == -1: 0.05; == 0: X1\_m')  
X2 <- recoder(X2, '== 1: 100; == -1: 50000; == 0: X2\_m')  
X3 <- recoder(X3, '== 1: 115600; == -1: 63070; == 0: X3\_m')  
X4 <- recoder(X4, '== 1: 900; == -1: 1110; == 0: X4\_m')  
X5 <- recoder(X5, '== 1: 63.1; == -1: 116; == 0: X5\_m')  
X7 <- recoder(X7, '== 1: 1680; == -1: 1120; == 0: X7\_m')  
  
num\_rows <- nrow(xd$design)  
  
X6 <- rep(X6\_m, num\_rows)  
X8 <- rep(X8\_m, num\_rows)  
  
mtxDopt <- matrix(c(X1,X2,X3,X4,X5,X6,X7,X8), ncol = 8)

test\_mtx <- data.frame(cbind(runif(50, 0.05, 0.15),  
 runif(50, 100, 50000),  
 runif(50, 63070, 115600),  
 runif(50, 900, 1110),  
 runif(50, 63.1, 116),  
 runif(50, 700, 820),  
 runif(50, 1120, 1680),  
 runif(50, 9855, 12045),   
 ncol = 8,   
 nrow = 50))

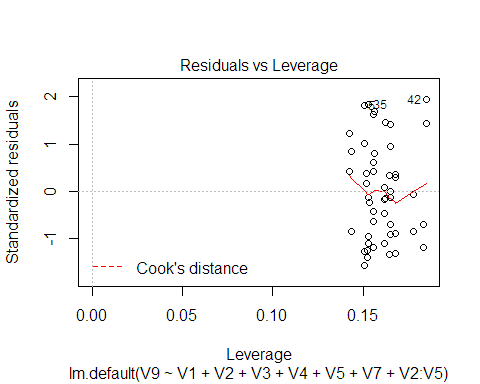
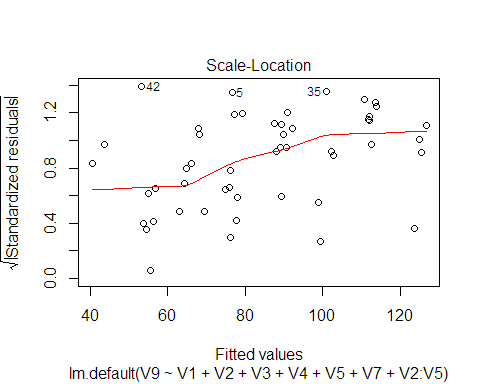
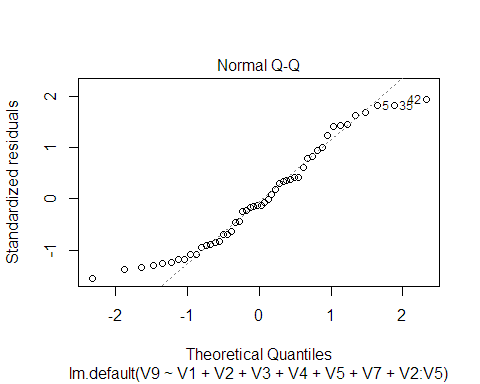
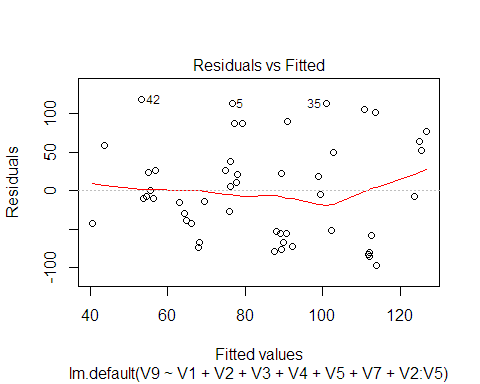
mtxRes2 <- matrix(nrow = num\_rows, ncol = 1)  
for (i in seq(1,50)){  
 mtxRes2[i,1] <- WaterFlowRateInBorehole(mtxDopt[i,])  
}  
mtx\_opt2 <- cbind(mtxDopt, mtxRes)  
eg2 <- as.data.frame(mtx\_opt2)  
  
optLm2 <- lm(V9~V1+V2+V3+V4+V5+V7+V2:V5, data = eg2)  
optPred2 <- predict(optLm2)  
summary(optLm2)

##   
## Call:  
## lm.default(formula = V9 ~ V1 + V2 + V3 + V4 + V5 + V7 + V2:V5,   
## data = eg2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -96.70 -54.42 -7.95 46.56 118.13   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 1.192e+02 1.300e+02 0.916 0.365  
## V1 -1.285e+02 1.916e+02 -0.671 0.506  
## V2 -4.476e-04 1.404e-03 -0.319 0.752  
## V3 -2.523e-05 3.700e-04 -0.068 0.946  
## V4 -6.299e-02 9.256e-02 -0.681 0.500  
## V5 6.545e-01 5.204e-01 1.258 0.216  
## V7 -3.163e-03 3.421e-02 -0.092 0.927  
## V2:V5 -7.877e-08 1.492e-05 -0.005 0.996  
##   
## Residual standard error: 67.58 on 42 degrees of freedom  
## Multiple R-squared: 0.1208, Adjusted R-squared: -0.02577   
## F-statistic: 0.8242 on 7 and 42 DF, p-value: 0.5729

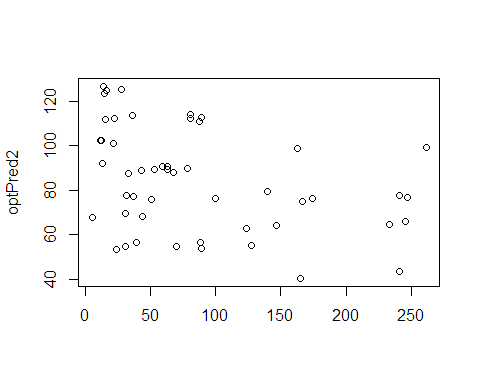
anova(optLm2)

## Analysis of Variance Table  
##   
## Response: V9  
## Df Sum Sq Mean Sq F value Pr(>F)   
## V1 1 2391 2390.9 0.5234 0.47339   
## V2 1 6463 6463.4 1.4150 0.24091   
## V3 1 3 3.1 0.0007 0.97950   
## V4 1 2629 2628.6 0.5755 0.45233   
## V5 1 14826 14826.4 3.2459 0.07878 .  
## V7 1 39 39.3 0.0086 0.92657   
## V2:V5 1 0 0.1 0.0000 0.99581   
## Residuals 42 191844 4567.7   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

plot(optLm2)



newMtx2 <- as.matrix(cbind(mtxRes2, optPred2))  
  
  
  
RMSEP <- function(y){  
 sqrt((sum((y[,2]-y[,1])^2))/50)  
}  
plot(newMtx2)



RMSEP(newMtx2)

## [1] 85.48002

chisq.test(newMtx2)

##   
## Pearson's Chi-squared test  
##   
## data: newMtx2  
## X-squared = 1839.2, df = 49, p-value < 2.2e-16

# Questions

1. Use an appropriate screening design to identify the significant input variables. Clearly specify the design used, the data simulated, subsequent analysis and significant variables. [10; Maximum of one page]
2. Based on the results of screening design develop a meta model for predicting WFR using a maximum of 50 runs (simulated values of WFR). Use D-optimal design and space filling design coupled with appropriate modelling methods. Compare and contrast the performance of the two models developed. Your response should include the designs used, search criterion, simulated data, metamodel and informative graphs for comparing the performance of two metamodesl.[25: Maximum of two page]

set.seed(13)  
lhs <- lhsDesign(50, 4, TRUE, seed = 13)  
full\_lhs <- lhsDesign(50, 8, TRUE, seed = 18)  
  
lhs\_d <- lhs$design  
flhs\_d <- full\_lhs$design

set.seed(13)  
ColX1 <- (0.05+(0.15-0.05)\*lhs\_d[,1])  
ColX2 <- (100+(50000-100)\*lhs\_d[,2])  
ColX4 <- (900+(1110-900)\*lhs\_d[,3])  
ColX5 <- (63.1+(116-63.1)\*lhs\_d[,4])  
ColX3 <- (89335)  
ColX6 <- (760)  
ColX7 <- (1400)  
ColX8 <- (10950)  
  
NewColX1 <- (0.05+(0.15-0.05)\*flhs\_d[,1])  
NewColX2 <- (100+(50000-100)\*flhs\_d[,2])  
NewColX3 <- (63070+(115600-63070)\*flhs\_d[,3])  
NewColX4 <- (900+(1110-900)\*flhs\_d[,4])  
NewColX5 <- (63.1+(116-63.1)\*flhs\_d[,5])  
NewColX6 <- (700+(820-700)\*flhs\_d[,6])  
NewColX7 <- (1120+(1680-1120)\*flhs\_d[,7])  
NewColX8 <- (9855+(12045-9855)\*flhs\_d[,8])  
  
  
  
X <- (cbind(ColX1, ColX2, ColX3, ColX4, ColX5, ColX6, ColX7, ColX8))  
X1 <-(cbind(NewColX1, NewColX2, NewColX3, NewColX4, NewColX5, NewColX6, NewColX7, NewColX8))  
X

## ColX1 ColX2 ColX3 ColX4 ColX5 ColX6 ColX7 ColX8  
## [1,] 0.08457936 21497.7557 89335 901.6480 115.87354 760 1400 10950  
## [2,] 0.14550773 4692.9707 89335 910.0646 91.31244 760 1400 10950  
## [3,] 0.12122073 19987.9055 89335 1089.9449 94.88280 760 1400 10950  
## [4,] 0.10581723 35032.1539 89335 1048.5860 114.31425 760 1400 10950  
## [5,] 0.05407587 49978.7137 89335 923.6731 72.25693 760 1400 10950  
## [6,] 0.11397813 17610.8691 89335 1009.1902 98.63475 760 1400 10950  
## [7,] 0.09885141 6941.0251 89335 970.1716 99.42017 760 1400 10950  
## [8,] 0.14647120 757.6795 89335 937.2656 106.77471 760 1400 10950  
## [9,] 0.08225324 8849.2037 89335 981.5109 77.19918 760 1400 10950  
## [10,] 0.13191787 18965.0580 89335 905.4599 86.49949 760 1400 10950  
## [11,] 0.07467776 32538.2690 89335 1078.9917 75.29907 760 1400 10950  
## [12,] 0.07624326 25476.0699 89335 1076.0101 64.31246 760 1400 10950  
## [13,] 0.11621888 20257.8789 89335 995.7587 93.12925 760 1400 10950  
## [14,] 0.05086744 39394.4489 89335 914.8255 65.41527 760 1400 10950  
## [15,] 0.13681291 23702.0237 89335 1106.6579 64.03509 760 1400 10950  
## [16,] 0.05727097 43559.5547 89335 974.8149 101.42269 760 1400 10950  
## [17,] 0.13328517 27745.3321 89335 1043.0896 78.94575 760 1400 10950  
## [18,] 0.14881709 22968.7050 89335 1024.0940 107.72865 760 1400 10950  
## [19,] 0.11026910 30646.4449 89335 1031.2167 82.23127 760 1400 10950  
## [20,] 0.08063895 1666.9597 89335 917.1177 111.84547 760 1400 10950  
## [21,] 0.06172587 12802.2015 89335 966.7651 109.15778 760 1400 10950  
## [22,] 0.05290610 15773.7849 89335 1042.2358 97.38713 760 1400 10950  
## [23,] 0.10264417 9711.1220 89335 1002.4075 80.39548 760 1400 10950  
## [24,] 0.09094384 29879.6986 89335 989.9846 100.90770 760 1400 10950  
## [25,] 0.11982648 47600.2399 89335 1066.1468 73.53246 760 1400 10950  
## [26,] 0.12476002 16132.3822 89335 1097.3301 89.54790 760 1400 10950  
## [27,] 0.10993655 38828.2578 89335 1021.0145 66.33173 760 1400 10950  
## [28,] 0.06707717 24586.9964 89335 954.3167 90.10204 760 1400 10950  
## [29,] 0.07133996 2895.8913 89335 1035.0532 81.25282 760 1400 10950  
## [30,] 0.09673296 46357.6847 89335 937.9800 110.30850 760 1400 10950  
## [31,] 0.06915378 26507.1170 89335 1054.6497 92.03843 760 1400 10950  
## [32,] 0.06318958 42564.5138 89335 955.1785 113.85579 760 1400 10950  
## [33,] 0.07816258 45065.8018 89335 949.2113 76.55155 760 1400 10950  
## [34,] 0.12638248 11653.0073 89335 1058.8672 110.75448 760 1400 10950  
## [35,] 0.12891982 44296.9576 89335 1085.3250 74.61906 760 1400 10950  
## [36,] 0.13978543 5497.7450 89335 1068.2432 70.72158 760 1400 10950  
## [37,] 0.08909111 40073.1053 89335 997.1307 104.27076 760 1400 10950  
## [38,] 0.14189607 28630.2562 89335 1099.2116 85.72970 760 1400 10950  
## [39,] 0.14248254 7764.9795 89335 961.2974 84.17709 760 1400 10950  
## [40,] 0.10183238 31038.6322 89335 945.3533 87.43869 760 1400 10950  
## [41,] 0.13479290 33798.5795 89335 987.6180 69.02571 760 1400 10950  
## [42,] 0.11554010 34094.5141 89335 1012.8980 67.92448 760 1400 10950  
## [43,] 0.07206633 13919.2324 89335 1103.8029 70.38940 760 1400 10950  
## [44,] 0.12279089 3165.8990 89335 1014.6075 105.38934 760 1400 10950  
## [45,] 0.09472079 37997.7298 89335 1083.5287 105.85943 760 1400 10950  
## [46,] 0.10616200 36704.4480 89335 976.0886 102.34384 760 1400 10950  
## [47,] 0.05970417 48394.0518 89335 1063.6815 96.39865 760 1400 10950  
## [48,] 0.06534175 14262.3215 89335 926.6210 94.20594 760 1400 10950  
## [49,] 0.08653514 41097.3542 89335 1027.9208 79.17718 760 1400 10950  
## [50,] 0.09263875 10233.8796 89335 930.7311 84.96128 760 1400 10950

set.seed(13)  
Xmat = X  
Xnew = X1  
  
lhd\_out <- matrix(nrow = 50, ncol = 1)  
for (i in seq(1,50)){  
 lhd\_out[i,1] <- WaterFlowRateInBorehole(X[i,])  
}  
  
y1 <- as.matrix(apply(X,1,WaterFlowRateInBorehole))  
y2 <- as.matrix(apply(Xnew,1,WaterFlowRateInBorehole))  
  
opt\_lhs <- as.matrix(cbind(X,y1))

set.seed(13)  
fit1 <- buildKriging(X, y1); fit1

## ------------------------  
## Forrester Kriging model.  
## ------------------------  
## Estimated activity parameters (theta) sorted   
## from most to least important variable   
## x3 x6 x7 x8 x1 x4 x5 x2   
## 1.154782 1.154782 1.154782 1.154782 0.5690379 0.1112051 0.03388143 1e-04  
##   
## exponent(s) p:  
## 2  
##   
## Estimated regularization constant (or nugget) lambda:  
## 0.03013262  
##   
## Number of Likelihood evaluations during MLE:  
## 456  
## ------------------------

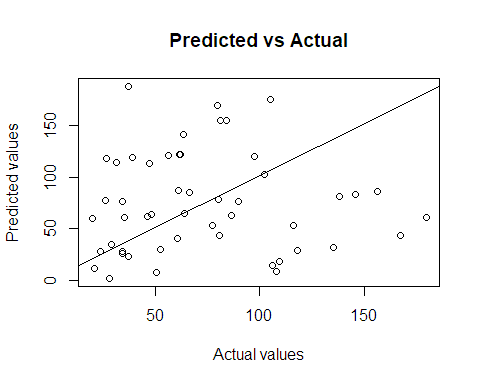
fit2 <- buildKriging(Xnew,y2); fit2

## ------------------------  
## Forrester Kriging model.  
## ------------------------  
## Estimated activity parameters (theta) sorted   
## from most to least important variable   
## x2 x1 x4 x6 x3 x5 x7 x8   
## 3.205895 1.993931 0.8057927 0.3113731 1e-04 1e-04 1e-04 1e-04  
##   
## exponent(s) p:  
## 2  
##   
## Estimated regularization constant (or nugget) lambda:  
## 0.1308744  
##   
## Number of Likelihood evaluations during MLE:  
## 988  
## ------------------------

hehe <- predict(fit1, X); hehe

## $y  
## [1] 32.36670 115.44088 115.83591 73.11907 20.45049 86.44894 57.99810  
## [8] 117.25566 46.06947 97.34660 46.13283 51.97466 90.00144 20.01293  
## [15] 164.35753 17.43804 135.91948 142.85423 91.70311 30.68252 18.83472  
## [22] 16.29397 75.36387 49.64872 119.43834 126.58830 96.53377 25.92346  
## [29] 36.45226 48.22098 32.20930 18.97042 39.17622 111.04522 139.72111  
## [36] 157.48617 46.97831 160.68371 127.84670 63.00136 129.63563 103.69907  
## [43] 46.65354 98.70133 63.96433 67.73756 22.44062 23.24916 55.69059  
## [50] 50.68885

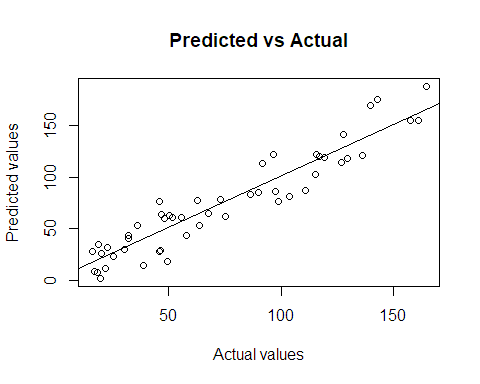
haha <- predict(fit2, Xnew)  
  
dOptRes <- as.matrix(cbind(haha$y,y1))  
  
dOptRes2 <- as.matrix(cbind(hehe$y,y1))  
  
plot(dOptRes,ylab = "Predicted values", xlab = "Actual values", main = "Predicted vs Actual")  
abline(1,1)



RMSEP(dOptRes)

## [1] 62.48371

RMSEP <- function(y){  
 sqrt((sum((y[,2]-y[,1])^2))/50)  
}  
  
plot(dOptRes2, ylab = "Predicted values", xlab = "Actual values", main = "Predicted vs Actual")  
abline(1,1)



RMSEP(dOptRes2)

## [1] 15.77823

1. Using best meta models in 2, and assuming that each range of input variable provided is the 5th, and 95th percentile respectively, generate the distribution of WFR. Present informative graphs and specify the features of the distribution. [10: Maximum of one page]

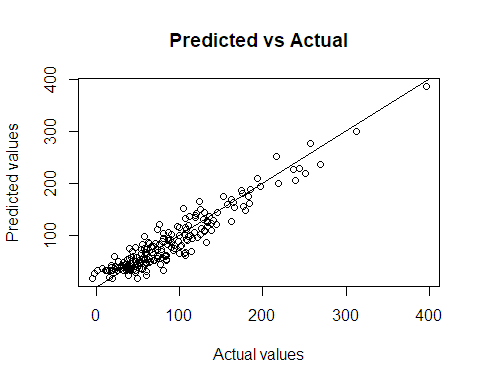
percentile <- qnorm(0.95)  
X1\_sd <- (X1\_m-0.05)/percentile  
X2\_sd <- (X2\_m-100)/percentile  
X4\_sd <- (X4\_m-900)/percentile  
X5\_sd <- (X5\_m-63.1)/percentile  
  
distMtx <- lhsDesign(200, 4, TRUE, seed = 63)  
matrice <- distMtx$design  
  
new\_Val1 <- abs(qnorm(matrice[,1], X1\_m, X1\_sd))  
new\_Val2 <- abs(qnorm(matrice[,1], X2\_m, X2\_sd))  
new\_Val4 <- abs(qnorm(matrice[,1], X4\_m, X4\_sd))  
new\_Val5 <- abs(qnorm(matrice[,1], X5\_m, X5\_sd))  
  
finalMtx <- matrix(cbind(new\_Val1, new\_Val2, X3\_m, new\_Val4, new\_Val5, X6\_m, X7\_m, X8\_m), ncol = 8)  
  
Pt3\_out <- matrix(nrow = 200, ncol = 1)  
for (i in seq(1,nrow(Pt3\_out))){  
 Pt3\_out[i,1] <- WaterFlowRateInBorehole(finalMtx[i,])  
}  
  
finalDf <- as.data.frame(cbind(finalMtx, Pt3\_out))  
is.factor(finalDf$V4)

## [1] FALSE

finalLm <- lm(V9~V1+V2+V4+V5+V1:V4, data= finalDf)  
summary(finalLm)

##   
## Call:  
## lm.default(formula = V9 ~ V1 + V2 + V4 + V5 + V1:V4, data = finalDf)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -47.024 -14.510 0.061 14.304 48.049   
##   
## Coefficients: (2 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.820e+01 1.112e+01 5.232 4.29e-07 \*\*\*  
## V1 -8.585e+03 6.749e+02 -12.720 < 2e-16 \*\*\*  
## V2 -1.351e-03 5.438e-04 -2.485 0.0138 \*   
## V4 NA NA NA NA   
## V5 NA NA NA NA   
## V1:V4 8.996e+00 6.602e-01 13.627 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 19.88 on 196 degrees of freedom  
## Multiple R-squared: 0.896, Adjusted R-squared: 0.8945   
## F-statistic: 563.2 on 3 and 196 DF, p-value: < 2.2e-16

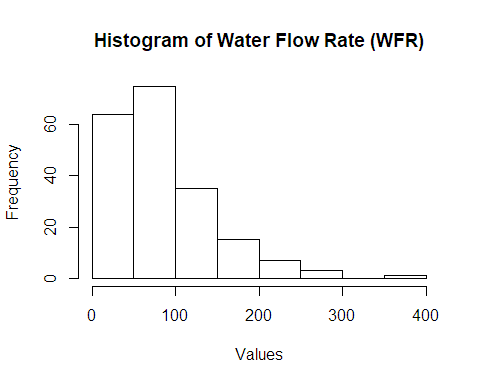
final\_Pred <- predict(finalLm)  
final\_final <- matrix(c(Pt3\_out, final\_Pred), ncol=2)  
  
plot(final\_final, ylab = "Predicted values", xlab = "Actual values", main = "Predicted vs Actual")  
abline(1,1)



RMSEP(final\_final)

## [1] 39.35845

hist(final\_Pred, main = "Histogram of Water Flow Rate (WFR)", xlab = "Values")

 distMtx <- lhsDesign(50, 4, TRUE) matrice <- distMtx$design

val\_X1 <- abs(qnorm(50, X1\_m, X1\_m/abs(sd))) val\_X2 <- abs(qnorm(50, X2\_m, X2\_m/abs(sd))) val\_X4 <- abs(qnorm(50, X4\_m, X4\_m/abs(sd))) val\_X5 <- abs(qnorm(50, X5\_m, X5\_m/abs(sd))) varMtx <- as.matrix(cbind(val\_X1,val\_X2,val\_X4,val\_X5), ncol = 4)

distMtx$design[,1]*varMtx[,1] newlyMtx <- matrix(nrow = 50, ncol = 4) mtxMult <- function(mtx1, mtx2){ for (i in seq(1,ncol(mtx1))){ newlyMtx[,i] <- mtx1[,i]*mtx2[,i] } return(newlyMtx) }

somMtx <- matrix(cbind(X3\_m,X6\_m,X7\_m,X8\_m), nrow=50, ncol = 4, byrow = TRUE) hello <- mtxMult(matrice,varMtx) Pt3df <- as.data.frame(cbind(hello,somMtx)) Pt3df <- as.matrix(Pt3df[c(“V1”,“V2”,“V5”,“V3”,“V4”,“V6”,“V7”, “V8”)]);Pt3df

Pt3\_out <- matrix(nrow = 50, ncol = 1) for (i in seq(1,50)){ Pt3\_out[i,1] <- WaterFlowRateInBorehole(Pt3df[i,]) }

finalMtx <- as.data.frame(cbind(Pt3df,Pt3\_out))

finalLm <- lm(V9~V1+V2+V3+V4+V1:V3, data= finalMtx) lm\_pred <- predict(finalLm) hist(lm\_pred) ```