Third Semester Project Paper Report

BY

Full Names: Anthony Imhenkuomon

Project Tittle: Fractional Factorial Design Using R Package RcmdrPlugin.DoE

Supervised by: Prof. Dr. Andreas Orth

IN

Department of High Integrity Systems

Frankfurt University of Applied Sciences.

July 2019

AIM

To carryout Fractional factorial 2-level Design of experiment (DoE) using RcmdrPlugings and to show a contour plot for the experiment.

Introduction

Fractional Factorial design is an experiment that explore the effect of different combination of factors on process output. Its is carried out as a subclass of all possible factors instead of complete set of all factors. This type of experimental design is very common most especially with 2- level factors. Fractional factorial designs are used when only some possible values are seen as relevant to the process model.

In R, there are several packages for creating and analyzing regular fractional factorial 2- level designs; However, : *FrF2* is the only package that relies on a all-inclusive catalogue of designs, automatically determines the overall best design or the best design for a certain estimation purpose (option estimable) and offers automatic blocking and automatic creation of split plot designs.

In this report, we are focusing on the installation process of *RcmdrPlugin.DoE* on R studio and how to navigate through the R commander GUI for creating various designs, statistical analysis and contour plots.

Starting R Commander and its DoE Plugin

For efficient utilization of R commander, the following packages need to be install first: *FrF2*, *DoE.base* provides the infrastructure for the suite and creates general factorial designs, *DoE.wrapper* interfaces to other packages for design of experiments on the Comprehensive R Archive Network and *RcmdrPlugin.DoE* provides graphical user interface (GUI) and FrF2.catlg128 supports *FrF2* for non-standard design creation tasks in 128 runs.

After a successful installation of these packages, to get started with the GUI within a new R session,

Enter the command:

require(RcmdrPlugin.DoE)

This opens up the R commander with its plugin Rcmdrplugin.DoE installed. This can be verified with the "Design" menu tab as seen below

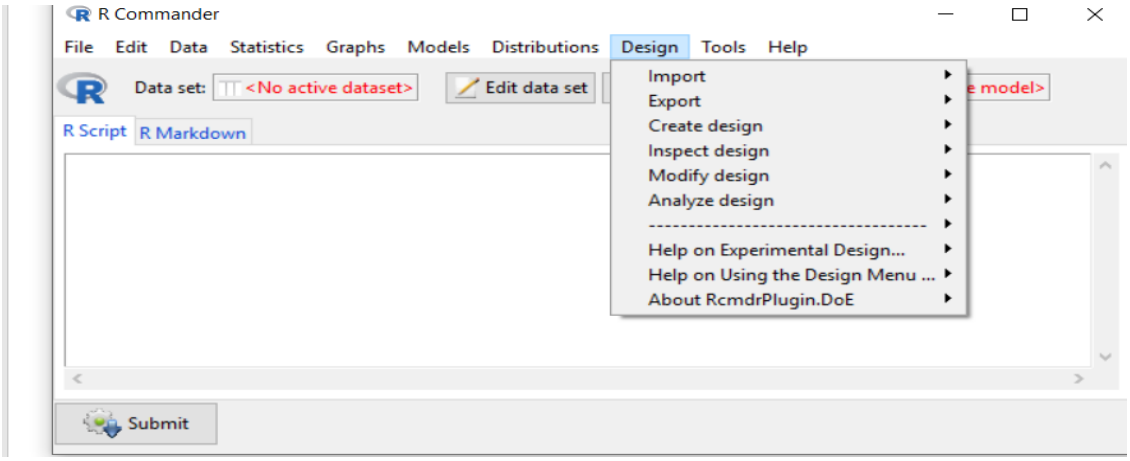


Figure 1: The R commander menu with expanded "Design" menu

There are a whole of functions that can be carried out with the GUI. Using the *Deofoamer* data set, we were able to display some of these functionalities.

Importing Data set

Data → import Data

Opens a dialog box as seen in figure.3. the user has the options of importing Dataset depends on the format of the data.

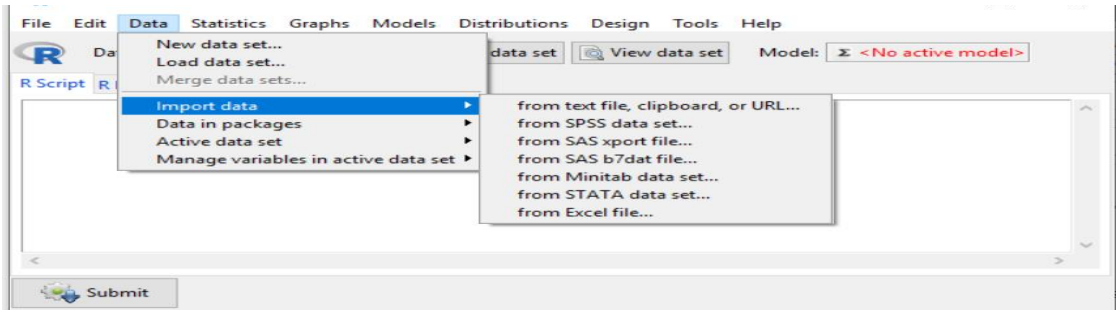


Figure 2. importing Dataset menu

After importing the dataset, we now have an active dataset "Dataset defoamer" as seen in figure 3. We can use the Edit menu to add, remove data by right clicking on the desired data column or row.

Creating Design

To open the Menu,

Design -> Create design -> Regular fractional Design..

A dialog box is opens up as in figure 4. For the Defoamer dataset, we use the regular fractional factorial. The procedure is similar to other design options. As seen from figure 5, we did not bother about number of Blocks and randomization just to keep it as simple as possible.

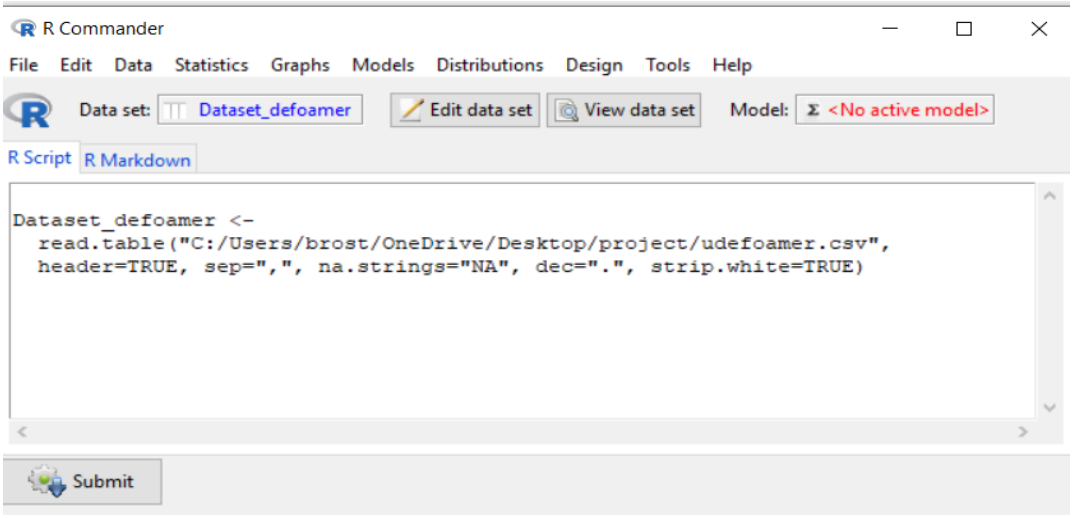


Figure 3. showing active imported Dataset "Dataset_defoamer"

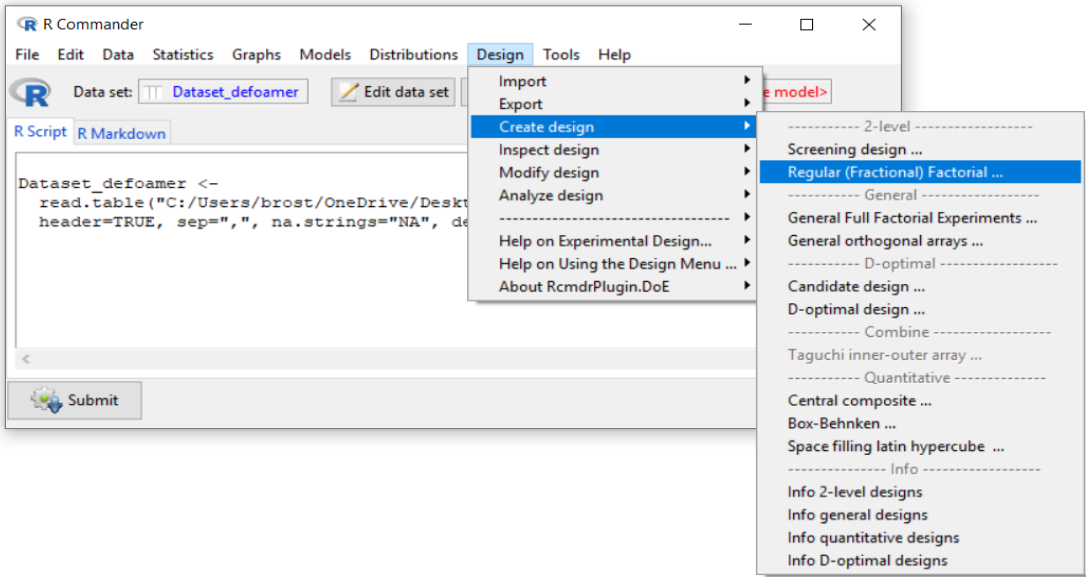


Figure 4: showing various design menu.

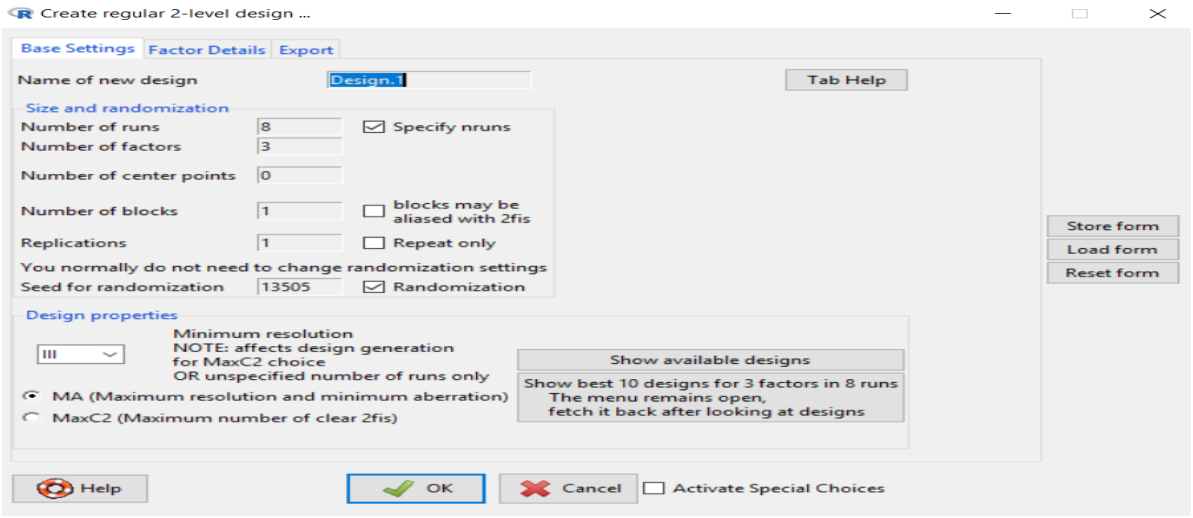


Figure 5. Frf2 with 3 factors 8 runs

In the factor details tab, we unchecked the "*common factor level"* in order to enter the factor variable names and values manually and to output the values in the design instead of the conventional *-1 and +1*.this is displayed in figure 6.

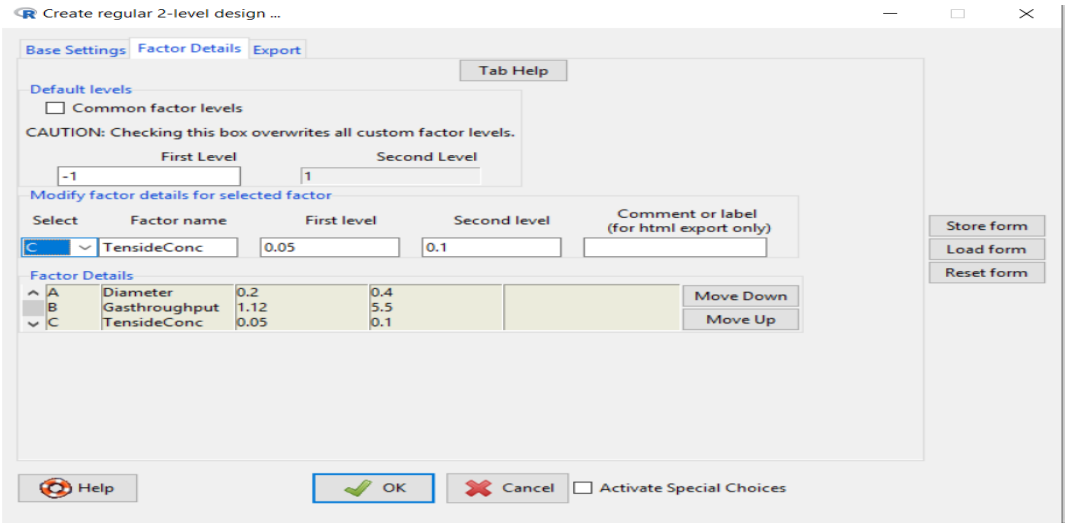


Figure 6:showing manually added factor details

To see the design after clicking OK button,

Design-> inspect Design-> display active design

Design-> inspect Design-> summarize active design

Hence, we generated the design below:

```
Rcmdr>  print( defoamerfrf2Design )
  Diameter GasThrougtput TensideConc
1      0.2           1.15         0.1
2      0.2            5.5        0.05
3      0.4           1.15        0.05
4      0.4           1.15         0.1
5      0.4            5.5         0.1
6      0.2            5.5         0.1
7      0.2           1.15        0.05
8      0.4            5.5        0.05
class=design, type= full factorial


Rcmdr>  summary( defoamerfrf2Design , brief = TRUE)
design was generated with RcmdrPlugin.DoE

Experimental design of type  full factorial
8  runs

Factor settings:
  Diameter GasThrougtput TensideConc
1      0.2           1.15        0.05
2      0.4            5.5         0.1
```

To add response variable to the design, in this case we used the *zvalue* data

Design-> modify Design-> add response variable(s).

Note that its possible to add more than one response variable. Figure 7 depict the above stated command. Select where your file is located. In our case we opt for *external csv file* and we gave it a name.

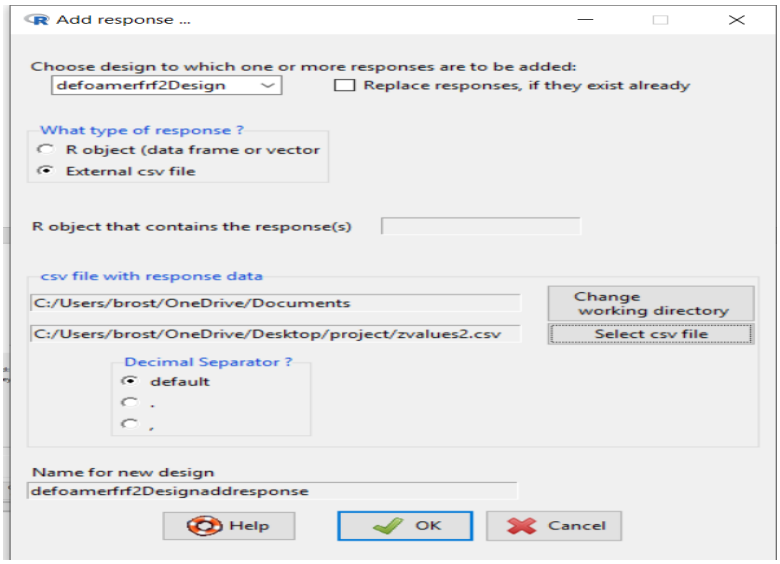


Figure 7. adding response variable dialog box

Data analysis

Main effect and Interaction plots

A **main effect** is the effect of a single independent variable on a dependent variable where all other independent variables are ignored.

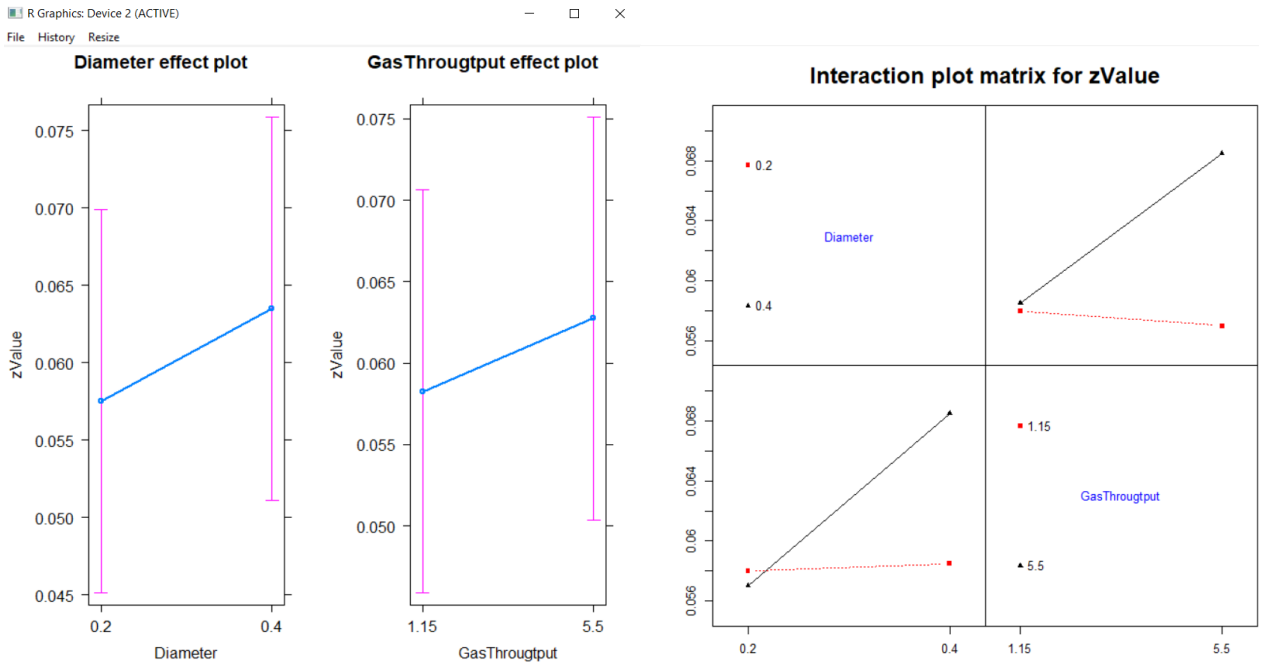
An interaction plot shows the levels of one variable on the X axis and has a separate line for the means of each level of the other variable. The Y axis is the dependent variable

Main effect and interaction plot are very important in analyzing Design of experiment (DoE). For the Defoamer dataset, we consider Diameter and GasThroughtput as factors with relevant effects on the response variable(zvalues) and a bit of interaction. a look at the *figure 8* connote the analysis.

Main effects and interaction plots for this design can be created from:

Design -> Analyze design ->Main effects and interaction plots

Plotting response means is one of the simplest analysis techniques but can be misleading in case of unbalanced data. It is generally safer – although neither foolproof – to plot effects from linear models. Hence, we use : *Models -> Graphs ->Effect plots* for the Main effect plot as seen below.



Statistical Analysis

one of the aim of this Project is to generate a contour plot for response variable as a function of two independent variable. Therefore, there is the need to carryout a linear model of the active dataset "defoamerfrfsadddrespone". From R commander GUI,

Statistics-> fit models-> linear Model

A dialog box will open as seen in figure 10. Enter the parameters as shown and click ok. To see the summary of the linear model,

Models-> summarize model

Below is a summary of our linear model with *zvalue* as response variable, *Diameter* and *GasThrougtput* as independent variable.

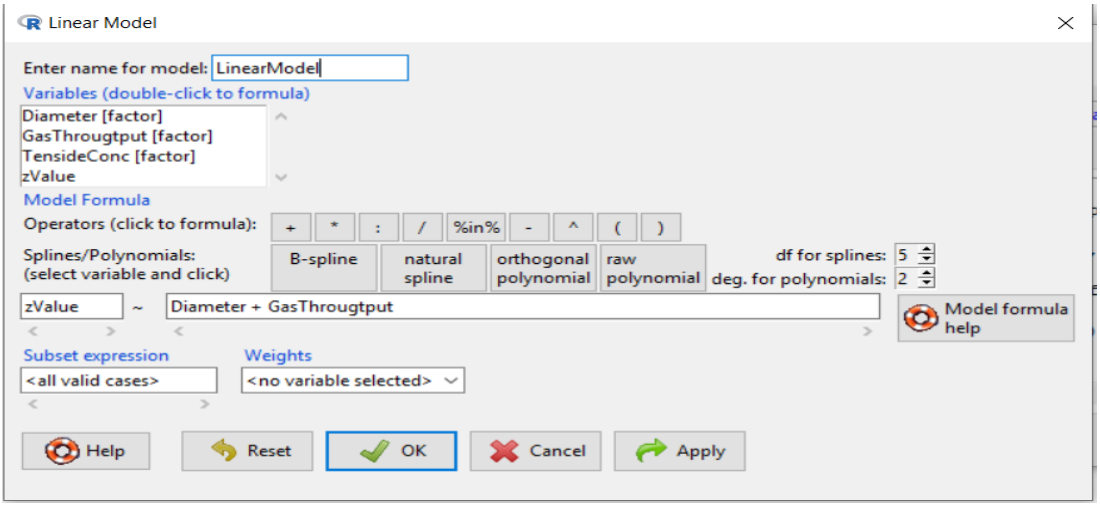


Figure 8.linear Model menu.

```
Rcmdr>  LinearModel <- lm(zValue ~ Diameter +GasThrougtput, data=defoamerfrf2Designaddresponse)

Rcmdr>  summary(LinearModel)

Call:
lm.default(formula = zValue ~ Diameter + GasThrougtput, data = defoamerfrf2Designaddresponse)

Residuals:
       1        2        3        4        5        6        7        8
-0.00925 -0.00775 -0.00425 -0.00125 -0.00275  0.00225  0.01475  0.00825

Coefficients:
              Estimate Std. Error t value  Pr(>|t|)
(Intercept)    0.060500   0.003404  17.773 0.0000103 ***
Diameter1      0.003000   0.003404   0.881     0.418
GasThrougtput1 0.002250   0.003404   0.661     0.538
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.009628 on 5 degrees of freedom
Multiple R-squared:  0.1953, Adjusted R-squared:  -0.1266
F-statistic: 0.6068 on 2 and 5 DF,  p-value: 0.5809
```

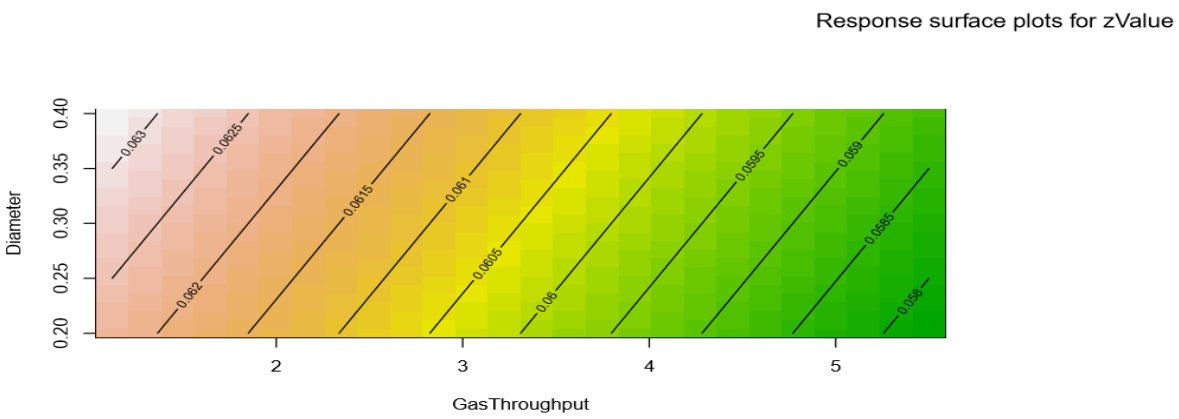
Contour Plot

Contour plots are used to show a three-dimensional surface on a two-dimensional plane. It plots two predic tor variables on the Y-axis and a response variable as contours.

Therefore, we want to deduce how response values(zvalue) changes as a function of input of Diameter, And GasThroughtput.

This can easily be done from the R commander GUI: with the Model still active,

Design-> Analyse Design-> Response surface plot->contour plot->select the paired independent variable s.



Conclusion

We have been able to generate Frf2 design, analyze the design with plots and Statistical model using linear model. Hence, we came up with a contour plot of the response variable as a function of the two most effect independent variable with the intercept 0.0605 at the center of the contour plot.

References

[1] Ulrika Grömping:“R Package FrF2 for creating and analyzing Fractional Factorial 2-Level Design”. Journal for Statistical software. January 2014, Volume 56, Issue 1

[2] Ulrikeá Grömping: “Tutorial for Designing Experiments using R package RcmdrPlugin.DoE

[3] https://www.statisticshowto.datasciencecentral.com/

Script

```
Dataset_defoamer <-
  read.table("C:/Users/brost/OneDrive/Desktop/project/udefoamer.csv",
  header=TRUE, sep=",", na.strings="NA", dec=".", strip.white=TRUE)
editDataset(Dataset_defoamer)
editDataset(Dataset_defoamer)
defoamerfrf2Design <- FrF2(nruns= 8 ,nfactors= 3 , blocks= 1 ,
  alias.block.2fis = FALSE , ncenter= 0 , MaxC2 = FALSE , resolution = NULL ,
  replications= 1 ,repeat.only= FALSE ,randomize= TRUE ,seed= 20219 ,
  factor.names=list( Diameter=c(0.2,0.4),GasThrougtput=c(1.15,5.5),
  TensideConc=c(0.05,0.1) ) )
## creator element of design.info will be different, when using the command line command!
print( defoamerfrf2Design )
summary( defoamerfrf2Design , brief = TRUE)
defoamerfrf2Designaddresponse                    <-                    add.response(defoamerfrf2Design,
"C:/Users/brost/OneDrive/Desktop/project/zvalues2.csv", replace=FALSE)
MEPlot(defoamerfrf2Designaddresponse, abbrev=4, select=c(1,2), response="zValue")
IAPlot(defoamerfrf2Designaddresponse, abbrev=4, show.alias=FALSE, select=c(1,2))
LinearModel <- lm(zValue ~ Diameter +GasThrougtput, data=defoamerfrf2Designaddresponse)
summary(LinearModel)
plot(allEffects(LinearModel))
par(mfrow=c( 2 , 1 ), oma=c(3,0,2,0))
contour( linearModel, as.list(c( ~Diameter*Gasthroughput )), image= TRUE  , atpos= 9 )
mtext( "Response surface plots for zValue" , outer=TRUE)
mtext( "Slice at:  Diameter=0.3, Gasthroughput=3.31" , outer=TRUE, side=1)
mtext( "Response surface plots for zValue" , outer=TRUE)
mtext( "Slice at:  Diameter=0.3, Gasthroughput=3.31" , outer=TRUE, side=1)
```