HW9

Richard Hardis, Alan Lo, Arjun Goyal, James Trawick

10/18/2019

## 12.1)

Design of Experiments is a frequently used technique in the field of Materials Science. In my undergraduate mat-sci lab, I was tasked with setting up a set of fatigue tests to determine several performance characteristics of a new nickel super-alloy. The alloy test samples were extremely expensive to acquire, so the number of tests that could be run was limited to 8. Of those 8, two were required for calibrating the test cell. With only 6 samples to test with and temperature level, corrosion level, and fatigue cycle tuning parameters to vary, the DOE approach was best suited for determining the parameter levels to use. Each of the three parameters was partitioned into discrete ranges and a fractional factorial DOE was used to determine the optimal set of parameter settings to best cover the entire design space of temperature, corrosion, and fatigue cycle tuning. Ideally, we would have had more samples to use because that would have provided better coverage of the design space, but using a DOE helped us maximize the usefulness of the testing period with only 6 samples to use.

## 12.2)

Here is the code used to answer 12.2:

library(FrF2)

## Loading required package: DoE.base

## Loading required package: grid

## Loading required package: conf.design

## Registered S3 method overwritten by 'DoE.base':  
## method from   
## factorize.factor 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

FrF2(16,10)

## A B C D E F G H J K  
## 1 -1 -1 -1 -1 1 1 1 1 -1 1  
## 2 -1 1 1 -1 -1 -1 1 1 -1 1  
## 3 1 1 1 1 1 1 1 1 1 1  
## 4 1 -1 1 1 -1 1 -1 1 -1 -1  
## 5 -1 -1 1 1 1 -1 -1 -1 -1 1  
## 6 1 -1 -1 -1 -1 -1 1 -1 -1 -1  
## 7 -1 1 -1 1 -1 1 -1 -1 -1 1  
## 8 1 -1 1 -1 -1 1 -1 -1 1 1  
## 9 1 1 -1 -1 1 -1 -1 -1 1 1  
## 10 1 1 -1 1 1 -1 -1 1 -1 -1  
## 11 -1 -1 1 -1 1 -1 -1 1 1 -1  
## 12 -1 1 1 1 -1 -1 1 -1 1 -1  
## 13 -1 1 -1 -1 -1 1 -1 1 1 -1  
## 14 1 -1 -1 1 -1 -1 1 1 1 1  
## 15 1 1 1 -1 1 1 1 -1 -1 -1  
## 16 -1 -1 -1 1 1 1 1 -1 1 -1  
## class=design, type= FrF2

The output of the FrF2 function effectively reduces the of the number of models required to analyze the performance of varying combinations of the yes/no features. Because there are no explicit labels for the features, we can say for example that if feature A is swimming pool, and B is back yard, and the associated values were 1 and -1 respectively for 1 of the 16 houses, then for that given house, include the swimming pool feature (A) and exclude the back yard feature (B). Above are the 16 experiments produced from running the FrF2 function on 10 features and 16 experiments.

## 14.1)

### 1)

For part 1, the mean and mode imputations for missing data were performed in MS Excel. See the attached Excel file in this zipped folder. For the mean imputation, the mean was computed for the column was calculated and used to fill each missing value. For the mode imputation the same technique was used except the column mode was used instead of column mean.

### 2)

For part 2, the missing values were imputed using regression. The missing value column was taken as the response variable and a simple linear regression model was built with all other columns as predictors. Each missing value was then imputed using the prediction supplied by the regression model using that columns specific predictor column values. See the included .R file for the code used to impute the values.

### 3)

For part 3, the same technique as part 2 was used, except a random value sampled from the mean 0, standard deviation 1 normal distribution was added to the output regression prediction as the perturbation. See the included .R file for the code used to impute the values.