SMDE SECOND ASSIGNMENT (60% OF THE FINAL MARK, INDIVIDUAL)
FIB-UPC June, 2021
Ali Arabyarmohammadi

GENERATE YOUR DATA.

The structure of the generated dataset must follow the structure presented on the next table. Here is not needed to use your RNG.

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	..	Factor 10	Answer
Individual 1								
Individual 2								
Individual 2000								

1. Define, for each factor (from 1 to 5) a distribution (the RVGs that you prefer, uniform, normal, exponential, etc.). For the factors 6 to 10 define a function that uses the previous variables, as an example F6=F1+2F3.
2. Define an answer variable that will be composed by a function that combines a subset of the previous factors plus a normal distribution you know (to add some random noise).

OBTAIN AN EXPRESSION TO GENERATE NEW DATA.

Imagine that you don't know nothing regarding how this dataset has been generated. **Consider that the factors represent different machines and the answer is the time to do an operation.**

You need to explore it because you want to define a model to obtain new data for your DOE (you want to detect the possible relations and the interactions between the factors, or maybe you want to test alternatives or predict future scenarios).

1. Explore the possible relations of all the factors and the answer variable, you can use any technique developed during the course (LRM or ANOVA).
2. Describe what you find on this analysis and, explain if it is coherent with the knowledge you have from the data.
3. Use a simulation model to generate new data. The simulation model will be a very simple model composed by one server by each one of the factors you use on the answer. If the answer is

   $answer = F3 + 5F6$ (+ Normal random noise).

   Then the model can be like the one represented below, where the arrivals will be represented by a constant distribution with 1 of value. The answer will be the overall service time.

SMDE SECOND ASSIGNMENT (60% OF THE FINAL MARK, INDIVIDUAL)
FIB-UPC June, 2021
Ali Arabyarmohammadi

DOE

Now you have a model to generate new data. This model can be used to generate data for the different scenarios that must be considered.

1. Define a DOE to explore with what parametrization of the 10 factors the answer obtains the best value (define what means best, i.e. maximize or minimize the value).
2. Detect and analyze the interactions.

Remember

- *Set the objectives.*
- *Select the process variables. Hypotheses to be tested, etc.*
- *Define an experimental design.*
- *Execute the design.*
- *Check that the data are consistent with the experimental assumptions.*
- *Analyze and interpret the results, detect effects of main factors and interactions.*

ANSWERS

Generating the dataset

To form our dataset, we are using our RNG to define a distribution for each factor F1 to F5 in R and for the factors F6 to F10, we will define a function that uses a combination of the variables F1 to F5. The dataset has 2000 rows and 10 factors. Then the *Answer* variable will be composed as an arbitrary subset of the previous variables and also adding some random noise.

$$answer = 2\text{ F8} + 3\text{ F3} + \text{F9} + \text{F5 (+ Normal random noise)}.$$

I have chosen 4 different distributions for the factor 1 to 5. Factor F1 and F5 are formed by Normal Distributions using mean=0, sd=1, and mean=1, sd=2 respectively. For the factors F2 I have used Exponential Distribution using rate=0.5, and also for the factor F3, a Uniform Distribution is used with the lower bound of 5 and the upper bound of 10. And finally, for factor F4, I have chosen a Beta distribution with the shape parameters 1 and 2. The real meaning of these factors is not very important but we can assume that they are representing the time needed to do a process by different machines and the *Answer* would be the time that is dependent on these factors.

Factor	Distribution	
F1	Normal Distribution	Mean: 0, Standard deviations: 1
F2	Exponential Distribution	rate=0.5
F3	Uniform Distribution	min=5, max=10
F4	Beta distribution	shape1 = 1, shape2 = 2
F5	Normal Distribution	Mean: 1, Standard deviations: 2
F6	F3 + 3 F1	
F7	F1 + F2	
F8	4 F5 + F3 + 2 F4	
F9	F5 + 2 F1	
F10	3 F3 + F1 + 3 F2	
Answer	2 F8 + 3 F3 + F9 + F5 + Normal distribution with mean **0** and standard deviations **1**	

By placing the factors 1 to 5 in the answer we can also define it regarding only these 5 factors. This answer variable could also be written as:

Answer = 2 F8 + 3 F3 + F9 + F5 + Normal distribution with mean 0 and standard deviations 1

Answer = 2 (4 F5 + F3 + 2 F4) + 3 F3 + (F5 + 2 F1) + F5 + Normal random noise

Answer = 10 F5 + 5 F3 + 4 F4 + 2 F1 + Normal random noise

This form of the ***answer*** will be useful when are obtaining a new answer in the LRM part.

Here is the R code that was written to generate the dataset:

```
# number of individuals (rows in the dataset)
n=2000

# Normal distribution
factor1 <- rnorm(n, mean=0, sd=1)

# Exponential Distribution
factor2 <- rexp(n, rate=.5)

# Uniform distribution
factor3 <- runif(n, min=5, max=10)

# Beta distribution
factor4 <- rbeta (n , 1 , 2)

# Normal distribution
factor5 <- rnorm(n, mean=1, sd=2)


# define function F6 to F10 that use the variables F1 to F5
factor6 <- factor3 +(3*factor1)
factor7 <- factor1 + factor2
factor8 <- (4*factor5) + factor3 + (2*factor4)
factor9 <-   factor5 +(2*factor1)
factor10 <- (3*factor3) + factor1 + (3*factor2)


err = rnorm(n, mean=0, sd=1)

# chosen answer variable
answer <-  (2*factor8) - (3*factor3) + factor9  - factor5 + err


# creating a data frame (Generating the dataset)
dset <- data.frame(factor1, factor2, factor3, factor4, factor5, factor6,
                   factor7, factor8, factor9, factor10, answer)

View (dset)
dim (dset)
str (dset)
describe (dset)
```

The dataset is visualized using a Scatterplot and a Boxplot and also shown in a paired panels.

```
#visualization dset
boxplot (dset, col="blue", border="brown")

pairs.panels (dset,
              method= "pearson",#correlation method
              hist.col= "#00AFBB",
              density= TRUE, #show density plot
              ellipses=TRUE, #show correlation ellipses
              pch=19)
```

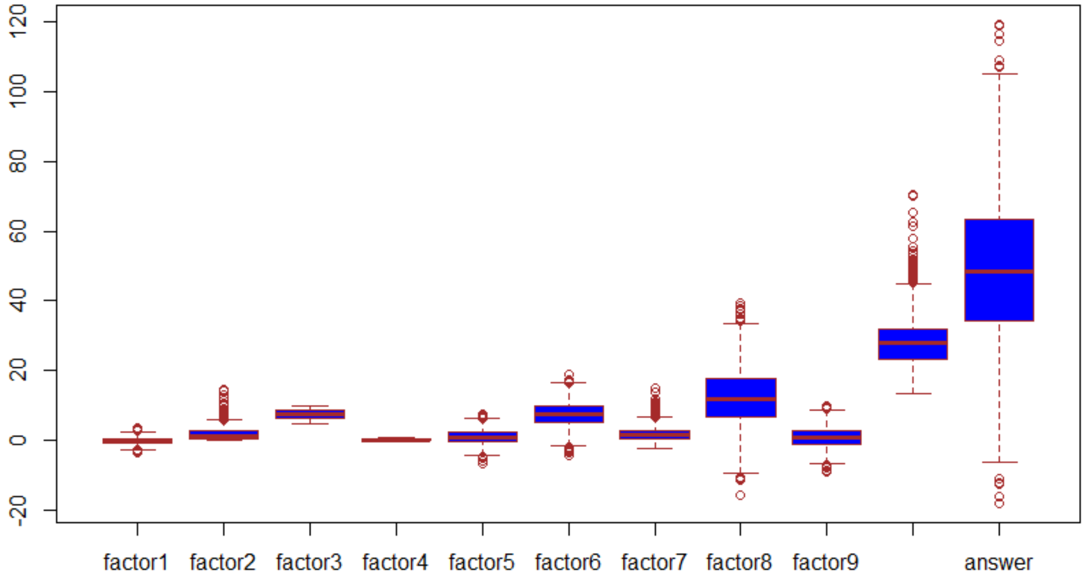


Figure 1 Visualization of the dataset using Boxplot

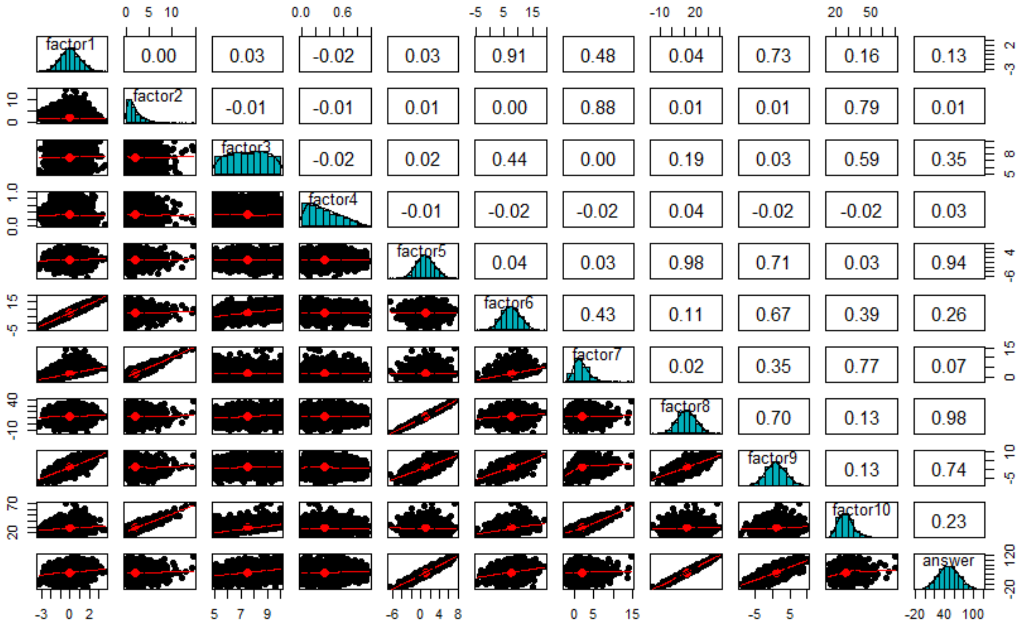


Figure 2 Visualization of the dataset in paired panels

```
> str (dset)
'data.frame':   2000 obs. of  11 variables:
 $ factor1 : num  0.37107 0.68267 1.25083 -0.00329 -0.01432 ...
 $ factor2 : num  0.9195 2.0447 0.0341 0.4426 1.4621 ...
 $ factor3 : num  6.37 7.41 9.19 8.89 7.35 ...
 $ factor4 : num  0.275 0.251 0.354 0.29 0.683 ...
 $ factor5 : num  -1.382 0.347 2.507 -0.243 -0.942 ...
 $ factor6 : num  7.49 9.46 12.95 8.88 7.3 ...
 $ factor7 : num  1.291 2.727 1.285 0.439 1.448 ...
 $ factor8 : num  1.4 9.3 19.93 8.49 4.95 ...
 $ factor9 : num  -0.639 1.712 5.009 -0.25 -0.97 ...
 $ factor10: num  22.3 29.1 28.9 28 26.4 ...
 $ answer  : num  18.4 43.9 75.7 41.6 30.8 ...
> describe (dset)
         vars    n  mean    sd median trimmed   mad    min    max  range  skew kurtosis   se
factor1     1 2000 -0.01  1.02  -0.02   -0.02  1.00  -3.30   3.69   7.00  0.04    -0.01 0.02
factor2     2 2000  1.95  1.88   1.38    1.64  1.37   0.00  14.51  14.51  1.89     5.14 0.04
factor3     3 2000  7.53  1.40   7.56    7.54  1.74   5.01  10.00   4.99 -0.04    -1.14 0.03
factor4     4 2000  0.33  0.23   0.29    0.31  0.26   0.00   0.96   0.96  0.58    -0.58 0.01
factor5     5 2000  1.01  2.00   0.94    0.99  1.95  -6.55   7.58  14.13  0.06     0.12 0.04
factor6     6 2000  7.50  3.40   7.45    7.48  3.41  -4.35  19.13  23.48  0.01    -0.02 0.08
factor7     7 2000  1.94  2.15   1.53    1.72  1.77  -2.41  14.86  17.26  1.33     3.30 0.05
factor8     8 2000 12.23  8.16  12.04   12.17  8.10 -15.63  39.58  55.21  0.07     0.03 0.18
factor9     9 2000  0.99  2.91   0.91    0.97  2.88  -9.08   9.95  19.03  0.06     0.04 0.06
factor10   10 2000 28.44  7.10  27.87   27.89  6.45  13.29  70.30  57.00  0.99     2.28 0.16
answer     11 2000 49.04 21.52  48.57   48.92 21.43 -17.85 119.14 136.98  0.07    -0.07 0.48
>
```

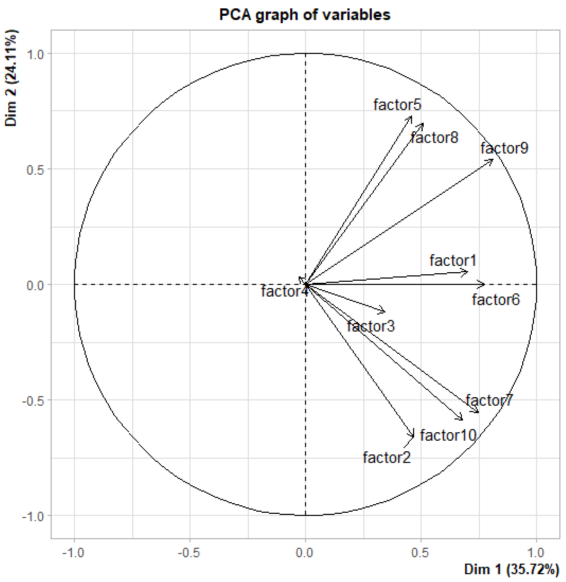
Obtaining an expression to generate new data

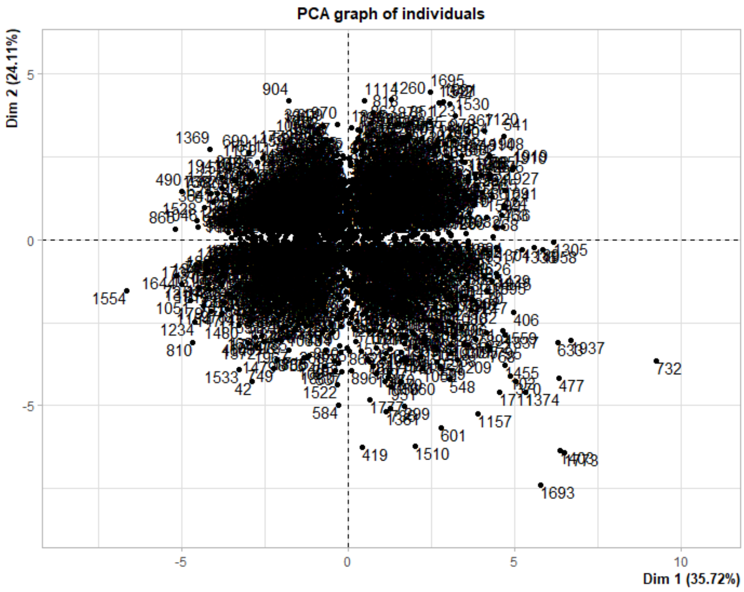
We will assume that we don't know anything regarding the dataset we generated in the previous part. We are going to define a model to obtain new data for our DOE. For this purpose, we will be exploring the dataset using the Multiple Linear Regression Model (MLR) and Principal Component Analysis (PCA).

PCA

Principal component analysis **(PCA)** is a technique for reducing the dimensionality of big datasets, increasing interpretability but at the same time minimizing information loss.
PCA transforms the original variables into a smaller set of linear combinations.
To explore the different relations and interactions between each variable, we do a PCA analysis. This analysis is used to define the main factors that exist on the dataset.
As we assume that we don't know anything about the dataset, we should consider all factors as active factors and remove the last column which is "answer" from the dataset. The output graphs contain a PCA **graph of individuals**, a PCA **graph of variables,** and a **Scree plot** which displays how much variation each principal component captures from the data.

```
#PCA analysis (Answer's column is removed)
pca1 = PCA(dset[,-11], scale.unit=TRUE, ncp =5,  graph=T)
pca1$eig
plot (pca1$eig[,1], type= "o" , main= "ScreePlot")
```






```
> pca1$eig
        eigenvalue percentage of variance cumulative percentage of variance
comp 1  3.572449e+00           3.572449e+01                          35.72449
comp 2  2.411478e+00           2.411478e+01                          59.83927
comp 3  1.729275e+00           1.729275e+01                          77.13203
comp 4  1.286631e+00           1.286631e+01                          89.99834
comp 5  1.000166e+00           1.000166e+01                         100.00000
comp 6  4.276669e-29           4.276669e-28                         100.00000
comp 7  4.810770e-30           4.810770e-29                         100.00000
comp 8  1.490619e-30           1.490619e-29                         100.00000
comp 9  6.593382e-32           6.593382e-31                         100.00000
comp 10 1.950890e-32           1.950890e-31                         100.00000
```

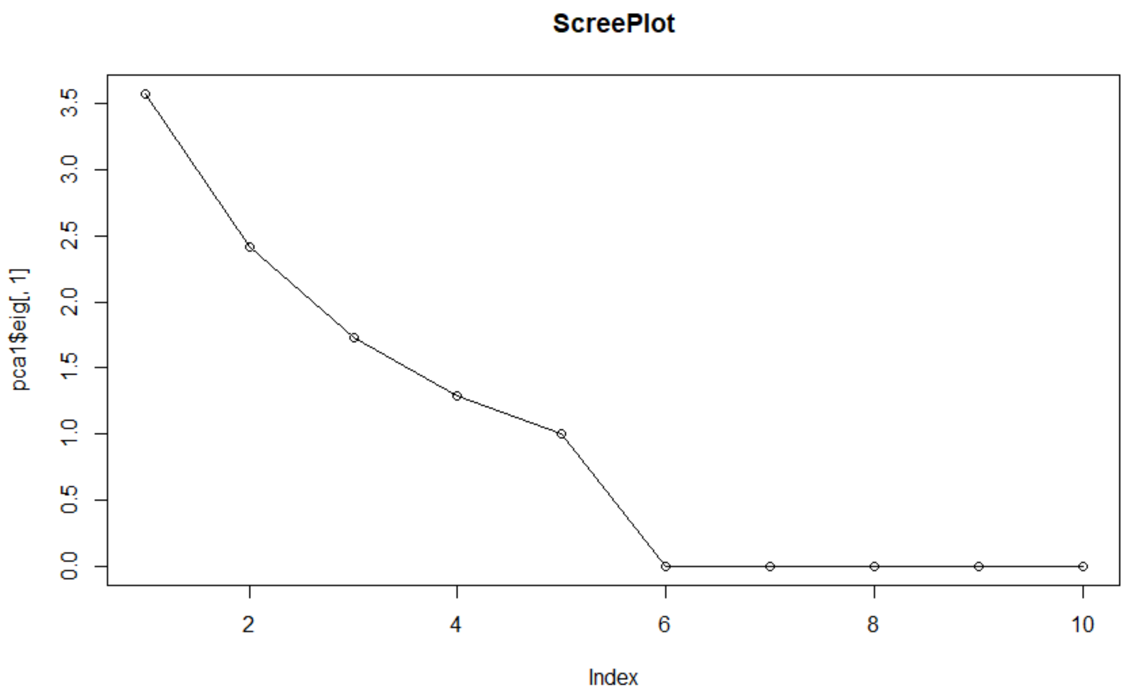


Figure 3 Amount of variation each principal component captures from the data

After examining the data, it is possible to conclude that there are two distinct sets of positively correlated variables. As we see in the variables factor map of PCA, by reducing the dimensions we almost lost 40.17 percent of the data but resume about 59.8 percent of the total variance of the dataset which is acceptable. After examining the data, it is possible to conclude that there are two distinct sets of positively correlated variables. The first group contains F5, F8 F9, and F1 and the second group contains F6, F3, F7, F10, and F2. We will continue exploring the possible relations between all variables of the dataset by performing several linear models.

Linear Regression Model

Multiple linear regression is used to estimate the relationship between **two or more independent variables** and **one dependent variable**.
We are going to apply a multiple linear regression model to the dataset that we generated in the beginning. The function ***lm*** in R is used for this purpose. Then we get the summary of the results.

The first linear model test was used to examine the relationship between the answer and all of the other factors in the dataset.

```
# Generating the linear model of the answer distribution
regmodel1 <- lm( answer ~ ., data=dset)
summary (regmodel1)
```

```
Call:
lm(formula = answer ~ ., data = dset)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0083 -0.6921 -0.0014  0.6798  3.5125 

Coefficients: (5 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.1318227  0.1275668   1.033    0.302    
factor1      2.0088118  0.0215910  93.039   <2e-16 ***
factor2     -0.0005355  0.0116914  -0.046    0.963    
factor3      4.9843000  0.0157155 317.158   <2e-16 ***
factor4      3.9777918  0.0948929  41.919   <2e-16 ***
factor5      9.9824981  0.0110239 905.532   <2e-16 ***
factor6             NA         NA      NA       NA    
factor7             NA         NA      NA       NA    
factor8             NA         NA      NA       NA    
factor9             NA         NA      NA       NA    
factor10            NA         NA      NA       NA    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9849 on 1994 degrees of freedom
Multiple R-squared:  0.9979, Adjusted R-squared:  0.9979 
F-statistic: 1.904e+05 on 5 and 1994 DF,  p-value: < 2.2e-16
```

We can observe from the results that a NA value is defined for F6 to F10. These NA items denote that each of these variables is a linear combination of the other variables, meaning that their behavior can be predicted using various combinations of the other factors. As a result, more tests will be required to determine their relationship. Furthermore, the p-value for F1, F3, F4, F5 is less than 0.05, while the t-value is significantly higher or lower than 0. As a result, we can conclude that these variables are related to the answer distribution. Instead, the findings reveal that factor2 is unrelated to answer distribution, so this variable like the factors 6 to 10 can be removed from the linear model.
The relationship between each element must be discovered before running the new linear model with the eliminated variables. To do so, a linear model is used for each of the variables that returned a NA result.

```
# Generating the linear model of factor6 distribution
regmodel2 <- lm(formula = factor6 ~ factor1 + factor2 + factor3 + factor4 + factor5,
                data = dset)
summary (regmodel2)
```

```
Call:
lm(formula = factor6 ~ factor1 + factor2 + factor3 + factor4 +
    factor5, data = dset)

Residuals:
       Min         1Q     Median         3Q        Max
-2.532e-13 -2.610e-16  1.280e-16  4.680e-16  8.024e-14

Coefficients:
              Estimate Std. Error    t value Pr(>|t|)
(Intercept) -1.568e-14  7.993e-16 -1.962e+01   <2e-16 ***
factor1      3.000e+00  1.353e-16  2.217e+16   <2e-16 ***
factor2     -3.030e-17  7.326e-17 -4.140e-01    0.679
factor3      1.000e+00  9.847e-17  1.016e+16   <2e-16 ***
factor4      1.709e-16  5.946e-16  2.870e-01    0.774
factor5      6.390e-17  6.908e-17  9.250e-01    0.355
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.172e-15 on 1994 degrees of freedom
Multiple R-squared:      1,	Adjusted R-squared:      1
F-statistic: 1.215e+32 on 5 and 1994 DF,  p-value: < 2.2e-16
```

We can observe from the data that factor1 and factor3 have p-values less than 0.05 and t-values much higher or lower than 0. As a result, we can conclude that these two factors are related to the factor6 distribution. The R-squared value in this example is 1, indicating that the confidence of correctly predicting the factor6 distribution in this generated linear model is very good. As a result, it is stated below.

Factor6: 3.0 Factor1 + 1.0 Factor3

```
# Generating the linear model of factor7 distribution
regmodel3 <- lm( formula = factor7 ~ factor1 + factor2 + factor3 + factor4 + factor5,
                 data = dset)
summary (regmodel3)
```

```
Call:
lm(formula = factor7 ~ factor1 + factor2 + factor3 + factor4 +
    factor5, data = dset)

Residuals:
       Min         1Q     Median         3Q        Max
-3.209e-15 -3.730e-16 -1.610e-16  4.800e-17  2.968e-13

Coefficients:
              Estimate Std. Error    t value Pr(>|t|)
(Intercept)  7.361e-15  8.625e-16  8.534e+00   <2e-16 ***
factor1      1.000e+00  1.460e-16  6.850e+15   <2e-16 ***
factor2      1.000e+00  7.905e-17  1.265e+16   <2e-16 ***
factor3     -8.688e-17  1.063e-16 -8.180e-01    0.414
factor4     -1.803e-16  6.416e-16 -2.810e-01    0.779
factor5     -9.457e-17  7.454e-17 -1.269e+00    0.205
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.66e-15 on 1994 degrees of freedom
Multiple R-squared:      1,	Adjusted R-squared:      1
F-statistic: 4.153e+31 on 5 and 1994 DF,  p-value: < 2.2e-16
```

We can observe from the data that factor1 and factor2 have p-values less than 0.05 and t-values much higher or lower than 0. As a result, we can conclude that these two factors are related to the factor7 distribution. The R-squared value in this example is 1, indicating that the confidence of correctly predicting the factor7 distribution in this generated linear model is very good. As a result, it is stated below.

Factor7: 1.0 Factor1 + 1.0 Factor2

```
# Generating the linear model of factor8 distribution
regmodel4 <- lm( formula = factor8 ~ factor1 + factor2 + factor3 + factor4 + factor5,
                 data = dset)
summary (regmodel4)
```

```
Call:
lm(formula = factor8 ~ factor1 + factor2 + factor3 + factor4 +
    factor5, data = dset)

Residuals:
       Min         1Q     Median         3Q        Max
-9.784e-14 -1.220e-15 -4.000e-16  5.400e-16  5.712e-13

Coefficients:
              Estimate Std. Error    t value Pr(>|t|)
(Intercept) -2.269e-15  1.760e-15 -1.289e+00    0.198
factor1      9.317e-17  2.979e-16  3.130e-01    0.754
factor2     -1.397e-16  1.613e-16 -8.660e-01    0.387
factor3      1.000e+00  2.168e-16  4.612e+15   <2e-16 ***
factor4      2.000e+00  1.309e-15  1.528e+15   <2e-16 ***
factor5      4.000e+00  1.521e-16  2.630e+16   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.359e-14 on 1994 degrees of freedom
Multiple R-squared:      1,	Adjusted R-squared:      1
F-statistic: 1.443e+32 on 5 and 1994 DF,  p-value: < 2.2e-16
```

We can observe from the data that factor3, factor4, and factor5 have p-values less than 0.05 and t-values much higher or lower than 0. As a result, we can conclude that these two factors are related to the factor8 distribution. The R-squared value in this example is 1, indicating that the confidence of correctly predicting the factor8 distribution in this generated linear model is very good. As a result, it is stated below.

Factor8: 4.0 Factor5 + 1.0 Factor3 + 2 Factor4

```
# Generating the linear model of factor9 distribution
regmodel5 <- lm( formula = factor9 ~ factor1 + factor2 + factor3 + factor4 + factor5,
                 data = dset)
summary (regmodel5)
```

```
Call:
lm(formula = factor9 ~ factor1 + factor2 + factor3 + factor4 +
    factor5, data = dset)

Residuals:
       Min         1Q     Median         3Q        Max
-1.896e-13 -1.000e-16  1.080e-16  3.140e-16  6.754e-14

Coefficients:
              Estimate Std. Error    t value Pr(>|t|)
(Intercept)  2.413e-15  6.370e-16  3.788e+00 0.000156 ***
factor1      2.000e+00  1.078e-16  1.855e+16  < 2e-16 ***
factor2     -3.462e-17  5.838e-17 -5.930e-01 0.553279
factor3      2.736e-17  7.847e-17  3.490e-01 0.727337
factor4     -3.316e-16  4.738e-16 -7.000e-01 0.484147
factor5      1.000e+00  5.504e-17  1.817e+16  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.918e-15 on 1994 degrees of freedom
Multiple R-squared:      1,	Adjusted R-squared:      1
F-statistic: 1.397e+32 on 5 and 1994 DF,  p-value: < 2.2e-16
```

We can observe from the data that factor1 and factor5 have p-values less than 0.05 and t-values much higher or lower than 0. As a result, we can conclude that these two factors are related to the factor8 distribution. The R-squared value in this example is 1, indicating that the confidence of correctly predicting the factor9 distribution in this generated linear model is very good. As a result, it is stated below.

Factor9: 2.0 Factor1 + 1.0 Factor5

```
# Generating the linear model of factor10 distribution
regmodel6<- lm( formula = factor10 ~ factor1 + factor2 + factor3 + factor4 + factor5,
                data = dset)
summary (regmodel6)
```

```
Call:
lm(formula = factor10 ~ factor1 + factor2 + factor3 + factor4 +
    factor5, data = dset)

Residuals:
       Min         1Q     Median         3Q        Max
-3.069e-12 -5.600e-16  1.530e-15  3.440e-15  4.641e-13

Coefficients:
              Estimate Std. Error    t value Pr(>|t|)
(Intercept) -1.315e-13  9.041e-15 -1.454e+01   <2e-16 ***
factor1      1.000e+00  1.530e-15  6.535e+14   <2e-16 ***
factor2      3.000e+00  8.286e-16  3.621e+15   <2e-16 ***
factor3      3.000e+00  1.114e-15  2.694e+15   <2e-16 ***
factor4      1.375e-15  6.725e-15  2.040e-01    0.838
factor5      7.982e-16  7.813e-16  1.022e+00    0.307
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.98e-14 on 1994 degrees of freedom
Multiple R-squared:      1, Adjusted R-squared:      1
F-statistic: 4.14e+30 on 5 and 1994 DF,  p-value: < 2.2e-16
```

We can observe from the data that factor1, factor2, and factor3 have p-values less than 0.05 and t-values much higher or lower than 0. As a result, we can conclude that these two factors are related to the factor8 distribution. The R-squared value in this example is 1, indicating that the confidence of correctly predicting the factor10 distribution in this generated linear model is very good. As a result, it is stated below.

Factor10: 1.0 Factor1 + 3.0 Factor2 + 3.0 Factor3

We can see that the answers produced are the same as what we defined first by comparing the linear combinations generated for all the factors from 6 to 10. Once determining all of the relationships between the variables with linear combinations, it's time to run the linear model of the answer without the discarded features. From the first LM we have done, we have found factor2 is unrelated to answer distribution and we won’t include it in the following regression.

```
# Generating the linear model without discarded factors
regmodel7 <- lm(formula = answer ~ factor1 + factor3 + factor4 + factor5    , data = dset)
summary (regmodel7)
```

```
Call:
lm(formula = answer ~ factor1 + factor3 + factor4 + factor5,
    data = dset)

Residuals:
    Min      1Q  Median      3Q     Max
-3.0085 -0.6926 -0.0026  0.6800  3.5111

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.13070    0.12516   1.044    0.296
factor1      2.00881    0.02159  93.063   <2e-16 ***
factor3      4.98431    0.01571 317.260   <2e-16 ***
factor4      3.97785    0.09486  41.934   <2e-16 ***
factor5      9.98249    0.01102 905.816   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9847 on 1995 degrees of freedom
Multiple R-squared:  0.9979, Adjusted R-squared:  0.9979
F-statistic: 2.381e+05 on 4 and 1995 DF,  p-value: < 2.2e-16
```

We can observe the all factors have p-values less than 0.05 and t-values much higher or lower than 0. As a result, we can conclude that all factors are related to the answer distribution. The R-squared value in this example is 0.9958, indicating that the confidence of correctly predicting the factor10 distribution in this generated linear model is very good. As a result, it is stated below.

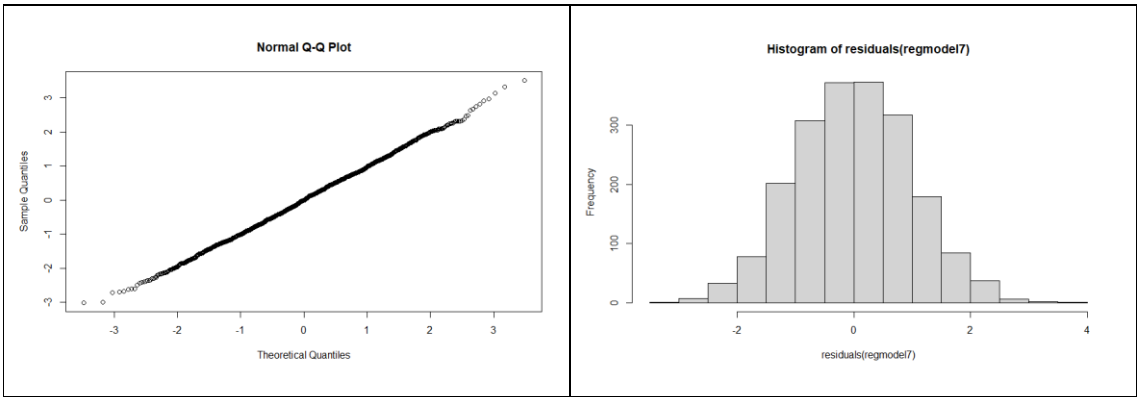
Lastly and before obtaining the answer expression, test assumptions for the linear model must be determined in order to show the validity of the test.

Testing the Assumptions of LRM

Normality test - Shapiro-Wilk normality test (and also using qqnorm and a histogram)

```
# Normality test
qqnorm(residuals(regmodel7))
hist(residuals(regmodel7))
```

Table 1 these graphs below show that the data is following a normal distribution




```
# Shapiro-Wilk normality test
shapiro.test(residuals(regmodel7))

        Shapiro-Wilk normality test

data:  residuals(regmodel7)
W = 0.99914, p-value = 0.4756
```

The calculated p-value is higher than the significance level of 0.05, so there is no indication that normality distribution is violated.

Homogeneity of variance - Breusch Pagan test

```
# Breusch Pagan test
lmtest::bptest(regmodel7)

        studentized Breusch-Pagan test

data:  regmodel7
BP = 4.2508, df = 4, p-value = 0.3731
```

The calculated p-value is higher than the significance level of 0.05, so there is no indication of heteroscedasticity on the calculated model.

Independence of variables - Durbin Watson test

```
# Durbin Watson test
lmtest::dwtest(regmodel7)
        Durbin-Watson test

data:  regmodel7
DW = 2.0233, p-value = 0.6988
alternative hypothesis: true autocorrelation is greater than 0
```

The calculated p-value is higher than the significance level 0, so the alternative hypothesis can be accepted. It means that the variables are not independent and it exits a correlation between them.

Obtaining a new answer

By confirming the validity of the test, we will be able to determine the following expression (y = b0 + b1*x1 + b2*x2 + b3*x3) that could be used to generate new data. (Numbers in red the number in green are marked in the last LRM results)

Answer = 0.13070 + ***10.00118 Factor5 + 4.99923 Factor3 + 4.00015 Factor4 + 2.01580 Factor1***

We can see the answer we obtained from the last LRM is the same as the answer we defined during defining our data set and the little difference between the numbers is due to the random noise we considered for the answer at the beginning.

Lastly, a prediction of the next value of the answer will be done to see if the linear model we just constructed is correct for our dataset. As a result, a new dataset with the same properties as the original will be created. This time the answer elements will be determined using the newly constructed linear model.

```
# Normal distribution
factor1 <- rnorm(50, mean=0, sd=1)

# Exponential Distribution
factor2 <- rexp(50, rate=.5)

# Uniform distribution
factor3 <- runif(50, min=5, max=10)

# Beta distribution
factor4 <- rbeta (50 , 1 , 2)

# Normal distribution
factor5 <- rnorm(50, mean=1, sd=2)


# define function F6 to F10 that use the variables F1 to F5
factor6 <-  factor3 +(3*factor1)
factor7 <-  factor1 + factor2
factor8 <-  (4*factor5) + factor3 + (2*factor4)
factor9 <-  factor5 +(2*factor1)
factor10 <- (3*factor3) + factor1 + (3*factor2)

# new answer variable
answer <-  0.13070 + (10.00118 * factor5) + (4.99923 * factor3) + (4.00015 * factor4) + (2.01580 * factor1)

dset2 <- data.frame(factor1, factor2, factor3, factor4, factor5, factor6,
                    factor7, factor8, factor9, factor10, answer)
```

Now after we formed our new dataset we will use the predict function in R to predict the next value of the answer.

```
#Predict with prediction interval
pred <- predict(regmodel7, newdata=dset2, interval="prediction")
pred

compData <- c()
predictedDataFrame <- data.frame(x1=pred)
answerDataFrame <- data.frame(x1=answer)
for(x in 1:nrow(predictedDataFrame)) {
  if( answerDataFrame$x1[x] >= predictedDataFrame$x1.lwr[x] &&
      answerDataFrame$x1[x] <= predictedDataFrame$x1.upr[x]){
    compData[x] <- TRUE
  }else{
    compData[x] <- FALSE
  }
}
predictedDataFrame["prediction"] <- compData
predictedDataFrame
```

Results

```
      x1.fit     x1.lwr     x1.upr  prediction
1   68.448117  66.515317  70.380918       TRUE
2   55.310356  53.378034  57.242679       TRUE
3   80.564417  78.629961  82.498873       TRUE
4    4.763249   2.828907   6.697591       TRUE
5   40.483506  38.550282  42.416731       TRUE
6   28.847041  26.914486  30.779596       TRUE
7   51.860504  49.927836  53.793172       TRUE
8   33.908932  31.975503  35.842361       TRUE
9   37.972410  36.038717  39.906104       TRUE
10  72.285945  70.353288  74.218601       TRUE
11  75.177914  73.244039  77.111789       TRUE
12  22.123975  20.190616  24.057333       TRUE
13 -17.669238 -19.605598 -15.732878       TRUE
14  17.111101  15.177671  19.044531       TRUE
15  42.534619  40.602350  44.466887       TRUE
16  38.196848  36.262400  40.131296       TRUE
17  25.828763  23.893038  27.764489       TRUE
18  19.044716  17.111390  20.978041       TRUE
19  58.723237  56.790632  60.655843       TRUE
20  44.566970  42.632129  46.501812       TRUE
21  62.637241  60.703679  64.570803       TRUE
22  50.222082  48.289053  52.155110       TRUE
23  57.524809  55.592604  59.457013       TRUE
24  20.423980  18.490709  22.357252       TRUE
25  34.593105  32.659407  36.526803       TRUE
26  39.874187  37.942205  41.806168       TRUE
27  46.282269  44.349946  48.214593       TRUE
28 -13.942889 -15.879549 -12.006229       TRUE
29  58.643836  56.712028  60.575644       TRUE
30  50.075969  48.142269  52.009670       TRUE
31  41.098569  39.166245  43.030892       TRUE
32  40.637209  38.704456  42.569962       TRUE
33  64.934049  63.000825  66.867272       TRUE
34  32.708332  30.775329  34.641336       TRUE
35  49.400833  47.463754  51.337912       TRUE
36  42.093574  40.161490  44.025657       TRUE
37  29.911811  27.978588  31.845035       TRUE
38  32.091365  30.158115  34.024615       TRUE
39  71.848644  69.914946  73.782343       TRUE
40  43.694449  41.759553  45.629344       TRUE
41  34.366535  32.432185  36.300886       TRUE
42  24.427510  22.493092  26.361927       TRUE
43  71.744959  69.811955  73.677962       TRUE
44  56.787200  54.855119  58.719280       TRUE
45  76.225861  74.292949  78.158773       TRUE
46  93.129058  91.194514  95.063601       TRUE
47  43.676702  41.744602  45.608802       TRUE
48  44.507244  42.575186  46.439302       TRUE
49  64.070991  62.137047  66.004935       TRUE
50  71.179390  69.246847  73.111932       TRUE
```

We got 50 True values in the result that are well-predicted values and as expected. The prediction interval obtained is then 100%, and it can be validated that the values produced are within the prediction interval of 95% of the samples, indicating that the model designed is accurate.

Simulation using GPSS

To simulate the data in **GPSS** we will be using the expression obtained from the Linear Regression Model. We should compare the ***answer*** obtained in GPSS with the ***answer*** in the corresponding row in the dataset. Time minus 1 in GPSS should be compared because the entity enters the system one second later than the beginning of the simulation. Here is a sample code snippet of the GPSS to simulate the model. (Random row in the dataset is selected to show the result)

```
; answer = 10.00118 * Factor5 + 4.99923 * Factor3 + 4.00015 * Factor4 + 2.01580 * Factor1

GENERATE 1
ADVANCE   (2.01580  # 1.63017987   +  ABS(Normal(1,0,1))) ;Factor1
ADVANCE   (4.99923  # 6.182544     +  ABS(Normal(1,0,1))) ;Factor3
ADVANCE   (4.00015  # 0.446788853  +  ABS(Normal(1,0,1))) ;Factor4
ADVANCE   (10.00118 # 1.558518000  +  ABS(Normal(1,0,1))) ;Factor5
TERMINATE 1
```

```
          GPSS World Simulation Report - Untitled Model 1.32.1


                 Wednesday, June 16, 2021 21:46:17

          START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
               0.000            54.682     6         0           0


 LABEL                 LOC  BLOCK TYPE     ENTRY COUNT CURRENT COUNT RETRY
                        1    GENERATE           54             0        0
                        2    ADVANCE            54             4        0
                        3    ADVANCE            50            31        0
                        4    ADVANCE            19             3        0
                        5    ADVANCE            16            15        0
                        6    TERMINATE           1             0        0
```

	factor1	factor2	factor3	factor4	factor5	factor6	factor7	factor8	factor9	factor10	answer
1	-0.29033934	7.943801e+00	8.261064	0.263721296	3.004368673	7.3900457	7.653461511	20.80598099	2.423689988	48.32425	71.507120
2	1.63017987	2.663852e-01	9.660620	0.691849232	1.630307372	14.5511593	1.896565106	17.56554765	4.890667117	31.41119	69.706010
3	0.32836868	4.464704e+00	9.188122	0.509045772	-0.744278707	10.1732277	4.793072671	7.22909834	-0.087541343	41.28685	41.474233
4	-0.60101252	2.234076e+00	7.368464	0.099828134	1.580458318	5.5654264	1.633063183	13.88995351	0.378433275	28.20661	50.963633
5	-1.56406737	2.422823e+00	9.001158	0.323819000	-1.211865946	4.3089556	0.858755526	4.80133192	-4.340000687	32.70787	31.823199
6	0.26643961	1.050962e+00	6.824901	0.732039781	2.266635029	7.6242202	1.317401841	17.35552101	2.799514256	23.89403	59.757650
7	1.88411265	8.440073e-01	7.140634	0.380350820	-1.210708656	12.7929719	2.728119922	3.05850094	2.557516638	25.83804	27.950847
8	1.16198402	1.925742e+00	6.182544	0.446788853	1.558518000	9.6684962	3.087725601	13.31019382	3.882486039	25.48684	53.056356
9	-0.87261698	1.200427e-04	8.673573	0.369997962	3.311490693	6.0557221	-0.872496938	22.65953174	1.566256732	25.14846	74.331093
10	-0.11333330	1.360227e+00	9.749958	0.562966963	1.091754615	9.4099584	1.246893496	15.24291067	0.865088020	33.21722	61.806595
11	-0.71339685	4.941632e-01	5.009642	0.464565012	-1.979733335	2.8694518	-0.219233616	-1.98016097	-3.406527033	15.79802	5.083103
12	0.51402568	1.391552e+00	7.489033	0.088367200	3.424626814	9.0311098	1.905578091	21.36427445	4.452678181	27.15578	73.899002

We see the **(time -1)** in GPSS is very close to our Answer in the dataset in highlighted row. This is just for showing a sample code and the results but next in the DOE section, we will do 10 replications and make a separate GPSS file for each scenario in the next part. (Please find GPSS files in the folder **"/GPSS_Replications**")

DOE

To define the relationship between factors affecting a process and the output of that process we use the Design of experiments (DOE). This systematic method is used to find cause-and-effect relationships and is needed to manage process inputs in order to minimize or maximize the output.

We'll use the previously created model data in this part to investigate how these records interact with the answer dependent variable. The goal of this final exercise is to conduct a design of tests with the model created in the previous session to see which parametrization of the variables yields the best result. The following statements must be defined to conduct an experiment design.

- ***Set the objectives.***

Specify the factor parametrization with which the answer yields the maximum value.

- ***Select the process variables. Hypotheses to be tested, etc.***

Examining the variables required considering factors 1 to 10 but, based on the findings of the preceding exercise, it was discovered that the **answer** is only connected with **factor1, factor3, factor4**, and **factor5**.

So, with only these four parameters, a reduction can be achieved. The significance of this simplification will be discussed in the DOE section.

- ***Define an experimental design.***

A full factorial design must be undertaken in order to ensure that all of the combinations are investigated during the experimentation in order to identify the parametrization of the 10 factors with which the answer obtains the highest value. Because our factor values contain more than two levels, we need to perform $\boldsymbol{n} \times \boldsymbol{v}^{\boldsymbol{k}}$ experiments, where **n** is the number of replications, **v** is the number of levels for each factor, and **k** is the number of factors to be explored. The total number of experiments required with ten factors and more than two levels is excessive. The 2k experimental design is the best factorial design to achieve the goal, where k is the number of components being explored in the experiment.

Two-level factorial experiments are factorial experiments in which each factor is investigated at only two levels. The early stages of experimentation usually involve the investigation of a large number of potential factors to discover the "vital few" factors. Two-level factorial experiments are used during these stages to quickly filter out unwanted effects so that attention can then be focused on the important ones. All combinations of the levels of the factors are run with this design. The Yates algorithm will be used to simplify the experiment's interaction calculus.

Ten variables must be studied in the current experimental design. Then, for a single replication, a full factorial two-level experiment requires $2^{10} = 1024$ experiments. Hopefully, it was found during the last part that the answer distribution is only connected with **factor1**, **factor3**, **factor4**, and **factor5**. As a result, the number of factors can be reduced, and the number of runs required has been reduced to $\mathbf{2^4 = 16}$ experiments. Furthermore, replications are required in order to determine the impact of measurement errors but as replications are very time-consuming we will do 10 replications and in the **GPSS** we need to produce **160** separate files for each replication. Furthermore, replications are required in order to obtain information on measurement inaccuracies. Determining the number of replications required to produce a result within a particular confidence level. Lastly, $\mathbf{n} \times \mathbf{2}^{\boldsymbol{k}}$ is used to calculate the number of runs for this experiment, which is $\mathbf{10 \times 2^4}$**=160**. The **two** levels defined for each factor will be the **minimum** and the **maximum** value of each of them and **160** runs will be performed. (I have written a **VB.NET** code to read the excel files and generate **160 GPSS** files that are located in the project folder)

Note: There is a single excitable file which is located in the project folder called **ExcelToGPSS.exe**. This file is the compiled form of the VB.NET code.

- ***Execute the design.***

The dataset of the ten replications must be completed before it can be executed. Forming the datasets needed can be simplified with the factors 1, 3, 4, and 5 regarding the new expression found in the LRM part. Using a **for loop in R** we are going to generate the numbers needed for performing 10 replications. Only For the Normal distributions we use the absolute values to avoid negative numbers.

```
##################################
# 10 replications using a loop
##################################

#------------------------------------------------------------------
for (i in 1:10) {

# Define factor 2 to 5
factor1 <- rnorm(2000, mean=0, sd=1)
# Exponential Distribution
factor2 <- rexp(2000, rate=.5)
# Uniform distribution
factor3 <- runif(2000, min=5, max=10)
# Beta distribution
factor4 <- rbeta (2000 , 1 , 2)
# Normal distribution
factor5 <- rnorm(2000, mean=1, sd=2)
# Define the answer based on LRM
answer <-  0.13070 + (10.00118 * factor5) + (4.99923 * factor3) + (4.00015 * factor4) + (2.01580 * factor1)
# creating a data frame
data1Rep <- data.frame(factor1, factor3, factor4, factor5, answer)
# print the summary (To extract min and max of each factor)
print(summary(data1Rep))

min_Factor1 = min(abs(factor1))
max_Factor1 = max(abs(factor1))
min_Factor3 = min(factor3)
max_Factor3 = max(factor3)
min_Factor4 = min(factor4)
max_Factor4 = max(factor4)
min_Factor5 = min(abs(factor5))
max_Factor5 = max(abs(factor5))

cat("min_Factor1 =\t", min_Factor1,"\n")
cat("max_Factor1 =\t", max_Factor1,"\n")
cat("min_Factor3 =\t", min_Factor3,"\n")
cat("max_Factor3 =\t", max_Factor3,"\n")
cat("min_Factor4 =\t", min_Factor4,"\n")
cat("max_Factor4 =\t", max_Factor4,"\n")
cat("min_Factor5 =\t", min_Factor5,"\n")
cat("max_Factor5 =\t", max_Factor5,"\n")

cat("--------------------------------- ", "\n")
cat("--------------------------------- ", "\n")
cat("--------------------------------- ", "\n")
}
#------------------------------------------------------------------
```

We will read the max and min values from the summary of 10 datasets. (data1Rep, data2Rep)
The two levels for each factor must be identified when the required datasets have been generated. The maximum and minimum values will be applied for each of these levels, as described in the previous steps. As a result, these mins and the max numbers must be determined from the datasets that have been created.

```
    factor1             factor3           factor4             factor5            answer
 Min.   :-3.608408   Min.   : 5.001   Min.   :0.0006547   Min.   :-5.3344   Min.   :-16.83
 1st Qu.:-0.649812   1st Qu.: 6.258   1st Qu.:0.1407920   1st Qu.:-0.3459   1st Qu.: 34.00
 Median : 0.001081   Median : 7.483   Median :0.2885399   Median : 1.0901   Median : 49.52
 Mean   :-0.003610   Mean   : 7.494   Mean   :0.3285782   Mean   : 1.0420   Mean   : 49.32
 3rd Qu.: 0.665895   3rd Qu.: 8.717   3rd Qu.:0.4792253   3rd Qu.: 2.3336   3rd Qu.: 63.95
 Max.   : 3.574410   Max.   :10.000   Max.   :0.9894305   Max.   : 7.9626   Max.   :113.33
min_Factor1 =        0.0001037634
max_Factor1 =        3.608408
min_Factor3 =        5.001422
max_Factor3 =        9.99987
min_Factor4 =        0.0006547366
max_Factor4 =        0.9894305
min_Factor5 =        0.001767884
max_Factor5 =        7.962584
---------------------------------
---------------------------------
---------------------------------
    factor1             factor3           factor4             factor5            answer
 Min.   :-3.409207   Min.   : 5.000   Min.   :0.0001925   Min.   :-5.7010   Min.   :-30.74
```

```
 1st Qu.:-0.711249   1st Qu.: 6.179   1st Qu.:0.1357331   1st Qu.:-0.2894   1st Qu.: 34.49
 Median :-0.008889   Median : 7.517   Median :0.3126588   Median : 0.9199   Median : 48.07
 Mean   :-0.022218   Mean   : 7.480   Mean   :0.3441515   Mean   : 0.9533   Mean   : 48.39
 3rd Qu.: 0.661175   3rd Qu.: 8.696   3rd Qu.:0.5220244   3rd Qu.: 2.2425   3rd Qu.: 62.56
 Max.   : 3.197115   Max.   :10.000   Max.   :0.9897237   Max.   : 7.8002   Max.   :129.65
min_Factor1 =          0.0005637909
max_Factor1 =          3.409207
min_Factor3 =          5.000166
max_Factor3 =          9.999614
min_Factor4 =          0.0001925086
max_Factor4 =          0.9897237
min_Factor5 =          0.003472341
max_Factor5 =          7.800222
---------------------------------
---------------------------------
---------------------------------
    factor1             factor3          factor4            factor5            answer
 Min.   :-3.372026   Min.   :5.001   Min.   :0.000118   Min.   :-6.0955   Min.   :-34.50
 1st Qu.:-0.643181   1st Qu.:6.276   1st Qu.:0.131620   1st Qu.:-0.3222   1st Qu.: 34.96
 Median :-0.001215   Median :7.531   Median :0.293583   Median : 0.9764   Median : 49.31
 Mean   : 0.012554   Mean   :7.525   Mean   :0.335108   Mean   : 1.0123   Mean   : 49.24
 3rd Qu.: 0.655158   3rd Qu.:8.781   3rd Qu.:0.501974   3rd Qu.: 2.3725   3rd Qu.: 63.31
 Max.   : 3.895558   Max.   :9.999   Max.   :0.990895   Max.   : 7.1752   Max.   :113.84
min_Factor1 =          0.0007268499
max_Factor1 =          3.895558
min_Factor3 =          5.000561
max_Factor3 =          9.999421
min_Factor4 =          0.0001179996
max_Factor4 =          0.9908948
min_Factor5 =          0.000184324
max_Factor5 =          7.175217
---------------------------------
---------------------------------
---------------------------------
    factor1            factor3          factor4             factor5            answer
 Min.   :-3.64148   Min.   :5.000   Min.   :0.0002176   Min.   :-6.8377   Min.   :-32.79
 1st Qu.:-0.68942   1st Qu.:6.289   1st Qu.:0.1364312   1st Qu.:-0.4721   1st Qu.: 34.31
 Median : 0.03464   Median :7.510   Median :0.3065279   Median : 0.9583   Median : 48.81
 Mean   : 0.02023   Mean   :7.512   Mean   :0.3385150   Mean   : 0.9541   Mean   : 48.62
 3rd Qu.: 0.70930   3rd Qu.:8.786   3rd Qu.:0.5160299   3rd Qu.: 2.3435   3rd Qu.: 63.23
 Max.   : 3.00044   Max.   :9.999   Max.   :0.9907930   Max.   : 7.9183   Max.   :125.15
min_Factor1 =          0.0006879138
max_Factor1 =          3.641478
min_Factor3 =          5.000314
max_Factor3 =          9.999259
min_Factor4 =          0.0002175981
max_Factor4 =          0.990793
min_Factor5 =          0.001511877
max_Factor5 =          7.918329
---------------------------------
---------------------------------
---------------------------------
    factor1            factor3          factor4             factor5            answer
 Min.   :-3.80294   Min.   :5.002   Min.   :0.0005325   Min.   :-5.0729   Min.   :-22.90
 1st Qu.:-0.71713   1st Qu.:6.174   1st Qu.:0.1276577   1st Qu.:-0.3505   1st Qu.: 34.54
 Median :-0.03767   Median :7.385   Median :0.2810480   Median : 1.0397   Median : 48.82
 Mean   :-0.04679   Mean   :7.445   Mean   :0.3294701   Mean   : 1.0388   Mean   : 48.96
 3rd Qu.: 0.68105   3rd Qu.:8.629   3rd Qu.:0.5113273   3rd Qu.: 2.4122   3rd Qu.: 63.65
 Max.   : 3.31719   Max.   :9.998   Max.   :0.9823327   Max.   : 7.5662   Max.   :113.50
min_Factor1 =          3.460042e-05
max_Factor1 =          3.802939
min_Factor3 =          5.001835
max_Factor3 =          9.998252
min_Factor4 =          0.0005324765
max_Factor4 =          0.9823327
min_Factor5 =          0.0005762919
max_Factor5 =          7.566216
---------------------------------
---------------------------------
---------------------------------
    factor1            factor3          factor4             factor5            answer
 Min.   :-3.64287   Min.   :5.001   Min.   :0.0009798   Min.   :-5.6599   Min.   :-14.36
 1st Qu.:-0.66291   1st Qu.:6.228   1st Qu.:0.1342382   1st Qu.:-0.3668   1st Qu.: 34.55
 Median : 0.07194   Median :7.527   Median :0.2962541   Median : 1.0050   Median : 49.09
 Mean   : 0.03389   Mean   :7.526   Mean   :0.3359629   Mean   : 1.0146   Mean   : 49.32
 3rd Qu.: 0.74250   3rd Qu.:8.788   3rd Qu.:0.5116311   3rd Qu.: 2.3404   3rd Qu.: 64.64
 Max.   : 3.04673   Max.   :9.998   Max.   :0.9890679   Max.   : 7.7004   Max.   :119.84
min_Factor1 =          2.037343e-05
max_Factor1 =          3.642873
min_Factor3 =          5.000803
max_Factor3 =          9.998039
min_Factor4 =          0.000979788
max_Factor4 =          0.9890679
min_Factor5 =          0.000665678
max_Factor5 =          7.700372
---------------------------------
---------------------------------
---------------------------------
    factor1           factor3          factor4             factor5            answer
 Min.   :-3.6873   Min.   :5.006   Min.   :0.0000126   Min.   :-5.1188   Min.   :-20.72
 1st Qu.:-0.6960   1st Qu.:6.324   1st Qu.:0.1423792   1st Qu.:-0.3409   1st Qu.: 34.63
 Median :-0.0105   Median :7.534   Median :0.2890208   Median : 1.0227   Median : 49.55
 Mean   :-0.0110   Mean   :7.548   Mean   :0.3325910   Mean   : 1.0281   Mean   : 49.45
 3rd Qu.: 0.6712   3rd Qu.:8.794   3rd Qu.:0.4958816   3rd Qu.: 2.3560   3rd Qu.: 63.93
 Max.   : 3.2270   Max.   :9.999   Max.   :0.9920018   Max.   : 7.9113   Max.   :129.13
min_Factor1 =          0.0002917005
max_Factor1 =          3.687285
min_Factor3 =          5.005938
max_Factor3 =          9.998564
```

```
min_Factor4 =          1.256288e-05
max_Factor4 =          0.9920018
min_Factor5 =          0.0005431999
max_Factor5 =          7.911268
----------------------------------
----------------------------------
----------------------------------
    factor1           factor3          factor4            factor5           answer
 Min.    :-3.55900   Min.    :5.001   Min.    :0.0000284   Min.    :-6.5692   Min.    :-20.23
 1st Qu.:-0.62139   1st Qu.:6.157   1st Qu.:0.1390748   1st Qu.:-0.2916   1st Qu.: 35.58
 Median : 0.02772   Median :7.439   Median :0.2844356   Median : 1.0779   Median : 49.42
 Mean   : 0.03791   Mean   :7.442   Mean   :0.3268047   Mean   : 1.0597   Mean   : 49.32
 3rd Qu.: 0.68090   3rd Qu.:8.682   3rd Qu.:0.4807441   3rd Qu.: 2.4027   3rd Qu.: 62.80
 Max.   : 2.98265   Max.   :9.999   Max.   :0.9550922   Max.   : 8.4139   Max.   :120.85
min_Factor1 =          0.0001350095
max_Factor1 =          3.558999
min_Factor3 =          5.000984
max_Factor3 =          9.998997
min_Factor4 =          2.835084e-05
max_Factor4 =          0.9550922
min_Factor5 =          0.002665675
max_Factor5 =          8.413858
----------------------------------
----------------------------------
----------------------------------
    factor1              factor3            factor4             factor5            answer
 Min.   :-3.487722   Min.    : 5.006   Min.    :0.0000976   Min.    :-5.3035   Min.    :-23.84
 1st Qu.:-0.670039   1st Qu.: 6.223   1st Qu.:0.1395444   1st Qu.:-0.3247   1st Qu.: 34.87
 Median : 0.025517   Median : 7.507   Median :0.2998153   Median : 1.0365   Median : 49.07
 Mean   : 0.006652   Mean   : 7.501   Mean   :0.3384988   Mean   : 1.0138   Mean   : 49.14
 3rd Qu.: 0.693193   3rd Qu.: 8.760   3rd Qu.:0.5130756   3rd Qu.: 2.4531   3rd Qu.: 64.54
 Max.   : 3.937069   Max.   :10.000   Max.   :0.9927728   Max.   : 7.1483   Max.   :119.77
min_Factor1 =          9.169633e-05
max_Factor1 =          3.937069
min_Factor3 =          5.005781
max_Factor3 =          9.999624
min_Factor4 =          9.756719e-05
max_Factor4 =          0.9927728
min_Factor5 =          6.530282e-07
max_Factor5 =          7.148334
----------------------------------
----------------------------------
----------------------------------
    factor1              factor3           factor4             factor5            answer
 Min.   :-3.606381   Min.    :5.003   Min.    :0.0000641   Min.    :-5.7658   Min.    :-24.40
 1st Qu.:-0.686325   1st Qu.:6.299   1st Qu.:0.1336889   1st Qu.:-0.4151   1st Qu.: 34.59
 Median :-0.037148   Median :7.544   Median :0.2803335   Median : 1.0268   Median : 49.36
 Mean   : 0.008639   Mean   :7.542   Mean   :0.3276715   Mean   : 0.9537   Mean   : 48.70
 3rd Qu.: 0.735519   3rd Qu.:8.786   3rd Qu.:0.4914276   3rd Qu.: 2.2807   3rd Qu.: 63.13
 Max.   : 3.968519   Max.   :9.999   Max.   :0.9684727   Max.   : 7.1082   Max.   :121.20
min_Factor1 =          0.0002070577
max_Factor1 =          3.968519
min_Factor3 =          5.0034
max_Factor3 =          9.999191
min_Factor4 =          6.411577e-05
max_Factor4 =          0.9684727
min_Factor5 =          0.003971868
max_Factor5 =          7.108243
----------------------------------
----------------------------------
```

Factor1	2.0158
Factor3	4.99923
Factor4	4.00015
Factor5	10.00118

The Min and max values of reach factor for each replications obtained from R

	Rep1	Rep2	Rep3	Rep4	Rep5	Rep6	Rep7	Rep8	Rep9	Rep10
min_Factor1	0.000103763	0.000563791	0.00072685	0.000687914	3.46E-05	2.04E-05	0.000291701	0.00013501	9.17E-05	0.000207058
max_Factor1	3.608408	3.409207	3.895558	3.641478	3.802939	3.642873	3.687285	3.558999	3.937069	3.968519
min_Factor3	5.001422	5.000166	5.000561	5.000314	5.001835	5.000803	5.005938	5.000984	5.005781	5.0034
max_Factor3	9.99987	9.999614	9.999421	9.999259	9.998252	9.998039	9.998564	9.998997	9.999624	9.999191
min_Factor4	0.000654737	0.000192509	0.000118	0.000217598	0.000532477	0.000979788	1.26E-05	2.84E-05	9.76E-05	6.41E-05
max_Factor4	0.9894305	0.9897237	0.9908948	0.990793	0.9823327	0.9890679	0.9920018	0.9550922	0.9927728	0.9684727
min_Factor5	0.001767884	0.003472341	0.000184324	0.001511877	0.000576292	0.000665678	0.0005432	0.002665675	6.53E-07	0.003971868
max_Factor5	7.962584	7.800222	7.175217	7.918329	7.566216	7.700372	7.911268	8.413858	7.148334	7.108243

We are going to perform 10 replications.
For each row in the table below, we have a GPSS file. (160 Files)

		Factor1	Factor3	Factor4	Factor5	Value_1	Value_3	Value_4	Value_5	Answer	Mean (10 replications)
1 Rep	1	-	-	-	-	0.000103763	5.001422	0.000654737	0.001767884	25.024	25.02
	2	+	-	-	-	3.608408	5.001422	0.000654737	0.001767884	32.297	32.51
	3	-	+	-	-	0.000103763	9.99987	0.000654737	0.001767884	50.012	50.00
	4	+	+	-	-	3.608408	9.99987	0.000654737	0.001767884	57.286	57.49
	5	-	-	+	-	0.000103763	5.001422	0.9894305	0.001767884	28.979	28.96
	6	+	-	+	-	3.608408	5.001422	0.9894305	0.001767884	36.253	36.45
	7	-	+	+	-	0.000103763	9.99987	0.9894305	0.001767884	53.967	53.94
	8	+	+	+	-	3.608408	9.99987	0.9894305	0.001767884	61.241	61.43
	9	-	-	-	+	0.000103763	5.001422	0.000654737	7.962584	104.641	101.72
	10	+	-	-	+	3.608408	5.001422	0.000654737	7.962584	111.915	109.21
	11	-	+	-	+	0.000103763	9.99987	0.000654737	7.962584	129.630	126.70
	12	+	+	-	+	3.608408	9.99987	0.000654737	7.962584	136.903	134.19
	13	-	-	+	+	0.000103763	5.001422	0.9894305	7.962584	108.597	105.66
	14	+	-	+	+	3.608408	5.001422	0.9894305	7.962584	115.870	113.15
	15	-	+	+	+	0.000103763	9.99987	0.9894305	7.962584	133.585	130.64
	16	+	+	+	+	3.608408	9.99987	0.9894305	7.962584	140.859	138.13
	17	-	-	-	-	0.000563791	5.000166	0.000192509	0.003472341	25.034	
	18	+	-	-	-	3.409207	5.000166	0.000192509	0.003472341	31.905	
	19	-	+	-	-	0.000563791	9.999614	0.000192509	0.003472341	50.027	
	20	+	+	-	-	3.409207	9.999614	0.000192509	0.003472341	56.898	
	21	-	-	+	-	0.000563791	5.000166	0.9897237	0.003472341	28.992	
	22	+	-	+	-	3.409207	5.000166	0.9897237	0.003472341	35.863	
	23	-	+	+	-	0.000563791	9.999614	0.9897237	0.003472341	53.985	

*Figure 4: 10 replications (**160** runs) in Excel (File: "\Excel Files\Replications.xlsx")*

In Excel, we generate the Yates table for the 10 replications using the maximum and minimum values of each of the factors (Extracted from results in R). The response is then determined using the extreme values for 10 replications. Finally, the mean of the ten obtained values is calculated, and this calculated value is what the Yates algorithm will utilize as a response.

The finalized table is copied to a new file called ***DataForYatesInR.xlsx***, which is then imported into R.

```
library("readxl")

# Select an Excel file manually
yatesTable <- read_excel(file.choose(),1)
```

```
> yatesTable
# A tibble: 16 x 5
   Factor1 Factor3 Factor4 Factor5 Values
   <chr>   <chr>   <chr>   <chr>    <dbl>
 1 -       -       -       -         25.0
 2 +       -       -       -         32.5
 3 -       +       -       -         50
 4 +       +       -       -         57.5
 5 -       -       +       -         29.0
 6 +       -       +       -         36.4
 7 -       +       +       -         53.9
 8 +       +       +       -         61.4
 9 -       -       -       +        102.
10 +       -       -       +        109.
11 -       +       -       +        127.
12 +       +       -       +        134.
13 -       -       +       +        106.
14 +       -       +       +        113.
15 -       +       +       +        131.
16 +       +       +       +        138.
```

- Then we will make an **ANOVA** model and then perform a **Yates algorithm** in Rstudio.

```
#Yates algorithm
anovaModel <- aov(Values~ Factor1*Factor3*Factor4*Factor5, data=yatesTable)
anovaModel
yatesModel <- yates.effects(anovaModel, data = yatesTable)
yatesModel
```

```
> anovaModel
Call:
   aov(formula = Values ~ Factor1 * Factor3 * Factor4 * Factor5,
    data = yatesTable)

Terms:
                 Factor1   Factor3   Factor4   Factor5 Factor1:Factor3 Factor1:Factor4 Factor3:Factor4 Factor1:Factor5
Sum of Squares   224.400  2496.002    62.094 23531.560           0.000           0.000           0.000           0.000
Deg. of Freedom        1         1         1         1               1               1               1               1
                Factor3:Factor5 Factor4:Factor5 Factor1:Factor3:Factor4 Factor1:Factor3:Factor5 Factor1:Factor4:Factor5
Sum of Squares            0.000           0.000                   0.000                   0.000                   0.000
Deg. of Freedom               1               1                       1                       1                       1
                Factor3:Factor4:Factor5 Factor1:Factor3:Factor4:Factor5
Sum of Squares                    0.000                           0.000
Deg. of Freedom                       1                               1
```

```
> yatesModel <- yates.effects(anovaModel, data = yatesTable)
> yatesModel
                Factor1                 Factor3                         Factor4
           7.490000e+00            2.498000e+01                    3.940000e+00
                Factor5         Factor1:Factor3                 Factor1:Factor4
           7.670000e+01            9.684010e-15                    2.043765e-15
        Factor3:Factor4         Factor1:Factor5                 Factor3:Factor5
          -3.877424e-15            6.226799e-15                    5.634680e-15
        Factor4:Factor5 Factor1:Factor3:Factor4         Factor1:Factor3:Factor5
           1.661753e-15           -7.277333e-15                   -1.146037e-14
Factor1:Factor4:Factor5 Factor3:Factor4:Factor5 Factor1:Factor3:Factor4:Factor5
           3.285305e-15           -5.004360e-15                    1.948262e-15
```

```
> meanData  <- (sum(yatesTable$values)/16)
> meanData
[1] 81.575
```

Ultimately, the mean of the various factors is determined in order to determine the calculus's primary effects.

- ***Check that the data are consistent with the experimental assumptions.***

We run an ANOVA test and the related assumptions to the data collected to show data consistency. We accept the assumptions of normality, homogeneity of variance, and independence because all tests pass with the experimental data. For **all 10 replications,** we performed the assumption tests.

```
#-----------------------------------------------------------------
for (i in 1:10) {

  # Define factor 2 to 5
  factor1 <- rnorm(2000, mean=0, sd=1)
  # Exponential Distribution
  factor2 <- rexp(2000, rate=.5)
  # Uniform distribution
  factor3 <- runif(2000, min=5, max=10)
  # Beta distribution
  factor4 <- rbeta (2000 , 1 , 2)
  # Normal distribution
  factor5 <- rnorm(2000, mean=1, sd=2)
  # Define the answer based on LRM
  answer <-  0.13070 + (10.00118 * factor5) + (4.99923 * factor3) + (4.00015 * factor4) + (2.01580 * factor1)
  # creating a data frame
  data1Rep <- data.frame(factor1, factor3, factor4, factor5, answer)
  # print the summary (To extract min and max of each factor)


  #ANOVA test for each replication
  anovaModel <- aov(answer~ factor1*factor3*factor4*factor5 , data=data1Rep)

  #Shapiro-Wilk normality test
  print(shapiro.test(residuals(anovaModel)))


  #Breusch Pagan test
 print(lmtest::bptest(anovaModel))


  #Durbin Watson test
  print(lmtest::dwtest(anovaModel))

}
#-----------------------------------------------------------------
```

```
          Shapiro-Wilk normality test

data:  residuals(anovaModel)
W = 0.034562, p-value < 2.2e-16


          studentized Breusch-Pagan test

data:  anovaModel
BP = 5.5613, df = 15, p-value = 0.9862


          Durbin-Watson test

data:  anovaModel
DW = 1.1155, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0


          Shapiro-Wilk normality test

data:  residuals(anovaModel)
W = 0.31817, p-value < 2.2e-16


          studentized Breusch-Pagan test

data:  anovaModel
BP = 11.267, df = 15, p-value = 0.7335
```

```
           Durbin-Watson test

data:  anovaModel
DW = 0.986, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0


           Shapiro-Wilk normality test

data:  residuals(anovaModel)
W = 0.078542, p-value < 2.2e-16


           studentized Breusch-Pagan test

data:  anovaModel
BP = 9.4417, df = 15, p-value = 0.8533


           Durbin-Watson test

data:  anovaModel
DW = 1.6077, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0


           Shapiro-Wilk normality test

data:  residuals(anovaModel)
W = 0.034162, p-value < 2.2e-16


           studentized Breusch-Pagan test

data:  anovaModel
BP = 9.2451, df = 15, p-value = 0.8644


           Durbin-Watson test

data:  anovaModel
DW = 1.0889, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0


           Shapiro-Wilk normality test

data:  residuals(anovaModel)
W = 0.061433, p-value < 2.2e-16


           studentized Breusch-Pagan test

data:  anovaModel
BP = 5.6008, df = 15, p-value = 0.9857


           Durbin-Watson test

data:  anovaModel
DW = 1.4556, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0


           Shapiro-Wilk normality test

data:  residuals(anovaModel)
W = 0.053938, p-value < 2.2e-16


           studentized Breusch-Pagan test

data:  anovaModel
BP = 10.116, df = 15, p-value = 0.8124


           Durbin-Watson test

data:  anovaModel
DW = 1.3505, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0


           Shapiro-Wilk normality test

data:  residuals(anovaModel)
W = 0.058344, p-value < 2.2e-16


           studentized Breusch-Pagan test

data:  anovaModel
BP = 1.7998, df = 15, p-value = 1


           Durbin-Watson test

data:  anovaModel
DW = 0.99564, p-value < 2.2e-16
```

```
alternative hypothesis: true autocorrelation is greater than 0

           Shapiro-Wilk normality test

data:  residuals(anovaModel)
W = 0.099562, p-value < 2.2e-16

           studentized Breusch-Pagan test

data:  anovaModel
BP = 70.19, df = 15, p-value = 4.133e-09

           Durbin-Watson test

data:  anovaModel
DW = 1.2101, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0

           Shapiro-Wilk normality test

data:  residuals(anovaModel)
W = 0.073121, p-value < 2.2e-16

           studentized Breusch-Pagan test

data:  anovaModel
BP = 12.899, df = 15, p-value = 0.6101

           Durbin-Watson test

data:  anovaModel
DW = 1.4713, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0

           Shapiro-Wilk normality test

data:  residuals(anovaModel)
W = 0.044022, p-value < 2.2e-16

           studentized Breusch-Pagan test

data:  anovaModel
BP = 6.7936, df = 15, p-value = 0.9631

           Durbin-Watson test

data:  anovaModel
DW = 1.444, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0
```

We can see that all the assumptions of normality, homogeneity of variance, and independence for all 10 replications are passed.

- ***Analyze and interpret the results, detect effects of main factors and interactions***

The main goal of this experiment, as stated at the start of the Design of the experiment, is to identify the parametrization of the 10 variables with which the answer receives the maximum value. It means we want to know by changing which parameter we can optimize the answer. As a result, the factors we're looking for are those that have the most impact on the answer value. With the information gathered, any variables that have a negative impact will be removed. Because their relationship was previously evaluated, none of the criteria had a negative impact on the ***answer*** in this situation and there is no need to discard any factor.

```
> yatesModel
                  Factor1                    Factor3                          Factor4
             3.377340e+01               1.124094e+02                     1.769488e+01
                  Factor5            Factor1:Factor3                  Factor1:Factor4
             3.439029e+02              -2.865092e-14                    -5.928830e-14
          Factor3:Factor4            Factor1:Factor5                  Factor3:Factor5
            -3.560354e-14              -2.444878e-14                    -4.339659e-14
          Factor4:Factor5    Factor1:Factor3:Factor4          Factor1:Factor3:Factor5
            -5.478055e-14               2.085787e-14                     7.334635e-15
  Factor1:Factor4:Factor5    Factor3:Factor4:Factor5  Factor1:Factor3:Factor4:Factor5
             3.797202e-14               7.113068e-14                    -5.638500e-14
```

The value retrieved must be assessed for the factors and interactions that have a positive effect. There is no interaction with a strong effect, as we can see.

Considering that the calculated mean is 81.5, the following factors sorted by significance are the ones that are given the most weight:

1. **Factor5 (343.9)**
2. **Factor3 (112.4)**
3. **Factor1 (33.7)**
4. **Factor4 (17.69)**

Appendix

VB.NET code to generate GPSS files of 10 replications

Note: There is a single excitable file which is located in the project folder called ExcelToGPSS.exe. This file is the compiled form of the VB.NET code.

```
'  SMDE SECOND ASSIGNMENT (60% Of THE FINAL MARK, INDIVIDUAL)
'  FIB-UPC June, 2021
'  ALI ARABYARMOHAMMADI
'
'___
'
'________________________________________________________
'  In order to make the replications, for each row In the table of Excel file, we have a
GPSS file. (160 Files)

Imports System.IO
Imports System.Text
Imports Microsoft.Office.Interop.Excel
Imports Excel = Microsoft.Office.Interop.Excel

Public Class ExcelToGPSS

    Dim ListFactorsAndValues As New List(Of FactorsAndValues)

    Private Sub Button1_Click(ByVal sender As System.Object,
                              ByVal e As System.EventArgs) Handles Button1.Click


        Dim Mypath As String
        Dim OpenFileDialog1 As New OpenFileDialog
        OpenFileDialog1.Filter = "EXCEL FILE | *.xlsx"
        If OpenFileDialog1.ShowDialog = DialogResult.OK Then
            Mypath = OpenFileDialog1.FileName
        Else
            Exit Sub
        End If

        Dim InputFile = "Input File:  " & Mypath
        LabelInputFile.Text = InputFile
        Dim rootPath As String = Path.GetDirectoryName(Mypath) & "\GPSS_Replications\"
        LabelOutputFolder.Text = rootPath
        LabelWait.Visible = True






        ReadExcelAndFillTheList(Mypath)



        Dim folderPath = Path.Combine(rootPath, String.Format("{0}", Now.ToString("yyyy-
MM-dd_HH-mm-ss")))
        If Not Directory.Exists(folderPath) Then
            Directory.CreateDirectory(folderPath)
        End If



        Dim c As Integer = 0
        For Each item As FactorsAndValues In ListFactorsAndValues

            Dim stb As New StringBuilder
```

```
            Dim line1 = String.Format("{0,-5}{1,20}{2,20}{3,20}{4,20}", ";", "Factor1",
"Factor3", "Factor4", "Factor5")
            Dim line2 = String.Format("{0,-5}{1,20}{2,20}{3,20}{4,20}", ";",
item.SignFactor1.ToString.Trim(), item.SignFactor3.ToString.Trim(),
item.SignFactor4.ToString.Trim(), item.SignFactor5.ToString.Trim())

            Dim line3 = String.Format("{0,-5}{1,20}{2,20}{3,20}{4,20}", ";", "Value_1",
"Value_3", "Value_4", "Value5")
            Dim line4 = String.Format("{0,-5}{1,20}{2,20}{3,20}{4,20}", ";",
item.Value_1, item.Value_3, item.Value_4, item.Value_5)


            stb.AppendLine(line1)
            stb.AppendLine(line2)
            stb.AppendLine("; answer = 10.00118 * Factor5 + 4.99923 * Factor3 + 4.00015
* Factor4 + 2.01580 * Factor1")
            stb.AppendLine(vbNewLine)
            stb.AppendLine(vbNewLine)
            stb.AppendLine(vbNewLine)

            stb.AppendLine("GENERATE 1")
            stb.AppendLine("ADVANCE  (2.01580  # " & item.Value_1 & "  +
ABS(Normal(1,0,1))) ;Factor 1")
            stb.AppendLine("ADVANCE  (4.99923  # " & item.Value_3 & "  +
ABS(Normal(1,0,1))) ;Factor3")
            stb.AppendLine("ADVANCE  (4.00015  # " & item.Value_4 & "  +
ABS(Normal(1,0,1))) ;Factor4")
            stb.AppendLine("ADVANCE  (10.00118 # " & item.Value_5 & "  +
ABS(Normal(1,0,1))) ;Factor5")
            stb.AppendLine("TERMINATE 1")

            Dim srt As String
            If c <= 16 Then srt = "Rep1"
            If c > 16 And c <= 32 Then srt = "Rep2"
            If c > 32 And c <= 48 Then srt = "Rep3"
            If c > 48 And c <= 64 Then srt = "Rep4"
            If c > 64 And c <= 80 Then srt = "Rep5"
            If c > 80 And c <= 96 Then srt = "Rep6"
            If c > 96 And c <= 112 Then srt = "Rep7"
            If c > 112 And c <= 128 Then srt = "Rep8"
            If c > 128 And c <= 144 Then srt = "Rep9"
            If c > 144 And c <= 160 Then srt = "Rep10"


            Dim strFile As String = srt & "_" & c & ".gps"

            Dim path As String = folderPath & "\" & strFile


            Dim fs As FileStream = File.Create(path)
            fs.Close()
            Dim fileExists As Boolean = File.Exists(path)
            If fileExists = False Then Continue For


            Dim file1 As System.IO.StreamWriter
            file1 = My.Computer.FileSystem.OpenTextFileWriter(path, True)
            file1.Write(stb.ToString())
            file1.Close()

            c += 1
        Next




        LabelWait.Visible = False
        MsgBox("GPSS files have been successfully created!")
    End Sub

    Private Sub ReadExcelAndFillTheList(Mypath As String)
        Dim xlApp As Excel.Application

        xlApp = New Excel.ApplicationClass

        Dim xlWorkBook As Workbook
```

```
        Dim xlWorkSheet As Worksheet

        xlWorkBook = xlApp.Workbooks.Open(Mypath)
        xlWorkSheet = xlWorkBook.Worksheets("sheet1")
        'display the cells value B2



        For i As Integer = 0 To 159

            Dim FactorsAndValues As New FactorsAndValues

            FactorsAndValues.SignFactor1 = xlWorkSheet.Cells(17 + i, 5).value
            FactorsAndValues.SignFactor3 = xlWorkSheet.Cells(17 + i, 6).value
            FactorsAndValues.SignFactor4 = xlWorkSheet.Cells(17 + i, 7).value
            FactorsAndValues.SignFactor5 = xlWorkSheet.Cells(17 + i, 8).value



            FactorsAndValues.Value_1 = xlWorkSheet.Cells(17 + i, 9).value
            FactorsAndValues.Value_3 = xlWorkSheet.Cells(17 + i, 10).value
            FactorsAndValues.Value_4 = xlWorkSheet.Cells(17 + i, 11).value
            FactorsAndValues.Value_5 = xlWorkSheet.Cells(17 + i, 12).value


            ListFactorsAndValues.Add(FactorsAndValues)
        Next

        'edit the cell with new value

        xlWorkBook.Close()
        xlApp.Quit()

        releaseObject(xlApp)
        releaseObject(xlWorkBook)
        releaseObject(xlWorkSheet)
    End Sub

    Private Sub releaseObject(ByVal obj As Object)
        Try
            System.Runtime.InteropServices.Marshal.ReleaseComObject(obj)
            obj = Nothing
        Catch ex As Exception
            obj = Nothing
        Finally
            GC.Collect()
        End Try
    End Sub

    Private Sub ExcelToGPSS_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        LabelWait.Visible = False
    End Sub
End Class


Public Class FactorsAndValues
    Public Property SignFactor1 As String
    Public Property SignFactor3 As String
    Public Property SignFactor4 As String
    Public Property SignFactor5 As String
    Public Property Value_1 As Double
    Public Property Value_3 As Double
    Public Property Value_4 As Double
    Public Property Value_5 As Double


End Class
```