```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.2.2
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.2.2

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

Proposed design of the experiment

The aim of this experiment is to analyze an unknown model that takes 11 variables with values in the interval [-1, 1] as an input and produces a numerical output. By interacting with the system, I obtained 39 samples which can be seen in the file 'results_DOE.csv' or in my "playground".

I'll utilize multiple linear regression and ANOVA to identify the variables that best explain this model, as well as creating a model of my own.

Loading data

```
df <- read.csv("results_DOE.csv")
```

Data visualization

Let's take a first look at the data and check that it has been loaded in correctly.

```
df
```

```
##                   Date  x1  x2  x3  x4  x5  x6  x7  x8  x9 x10  x11          y
## 1  2023-02-13-14:40:55 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 0.1 0.11 -0.4452340
## 2  2023-02-13-14:41:21 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.00  1.0170653
## 3  2023-02-13-14:41:30 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.00  1.0164894
## 4  2023-02-13-14:41:37 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.00  1.0126852
## 5  2023-02-13-14:41:49 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.00 -0.9860999
## 6  2023-02-13-14:43:20 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 1.00 -0.9866759
## 7  2023-02-13-14:43:27 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.00 -0.9819591
## 8  2023-02-13-14:43:35 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 1.0 1.00  1.0135168
## 9  2023-02-13-14:43:40 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 1.0 1.00 -0.9886122
## 10 2023-02-13-14:43:48 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.00  1.0126511
## 11 2023-02-13-14:43:54 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.00  1.0123700
## 12 2023-02-13-14:44:08 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.00  1.0192621
## 13 2023-02-13-14:44:17 0.0 0.0 0.0 0.0 0.5 0.0 0.0 0.0 0.0 0.0 0.00  1.0180711
## 14 2023-02-13-14:44:23 0.0 0.0 0.0 0.0 0.2 0.0 0.0 0.0 0.0 0.0 0.00  1.0191879
## 15 2023-02-13-14:44:29 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.00  1.0182995
## 16 2023-02-13-14:44:33 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.00  2.0142335
## 17 2023-02-13-14:44:45 0.0 0.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.00  2.0151214
## 18 2023-02-13-14:44:49 0.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.00  2.0152756
## 19 2023-02-13-14:44:52 1.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.00  1.4266879
## 20 2023-02-13-14:45:00 1.0 1.0 1.0 1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.00  1.4187834
## 21 2023-02-13-14:45:11 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.00  1.0193823
## 22 2023-02-13-14:45:16 0.0 0.0 0.0 0.0 0.0 0.0 1.0 1.0 0.0 0.0 0.00  1.0154471
## 23 2023-02-13-14:45:24 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.00 -0.9851013
## 24 2023-02-13-14:45:29 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.00 -1.5826048
## 25 2023-02-13-14:45:34 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.00 -1.5812904
## 26 2023-02-13-14:45:39 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.00 -1.5839345
## 27 2023-02-13-14:45:44 1.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.00 -0.5822094
## 28 2023-02-13-14:45:54 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.00 -1.5749786
## 29 2023-02-13-14:45:58 1.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 1.0 0.0 0.00 -0.5775127
## 30 2023-02-13-14:46:11 1.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.00  1.4236339
## 31 2023-02-13-14:51:07 1.0 1.0 1.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.00  1.4251858
## 32 2023-02-13-14:51:20 1.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0 1.0 1.00  1.4192359
## 33 2023-02-13-14:51:31 1.0 1.0 1.0 1.0 0.0 0.0 0.0 1.0 0.0 1.0 1.00  1.4207786
## 34 2023-02-13-14:51:36 1.0 1.0 1.0 1.0 1.0 0.0 0.0 1.0 0.0 1.0 1.00  1.4170971
## 35 2023-02-13-14:51:41 1.0 1.0 1.0 1.0 1.0 1.0 0.0 1.0 0.0 1.0 1.00  1.4203601
## 36 2023-02-13-14:51:47 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.0 1.0 1.00  1.1252776
## 37 2023-02-13-14:51:53 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.00 -0.8816741
## 38 2023-02-13-14:51:58 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.0 1.00 -0.8791709
## 39 2023-02-13-14:52:03 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.0 0.00 -0.8826336
```

The "Date" variable is not relevant for our analysis, so let's remove it and see the summary of the data.

```
df <- subset(df, select = -Date)
summary(df)
```

```
##        x1               x2               x3               x4
##  Min.   :0.0000   Min.   :0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.000   Median :0.0000   Median :0.0000
##  Mean   :0.4641   Mean   :0.441   Mean   :0.4436   Mean   :0.4462
```

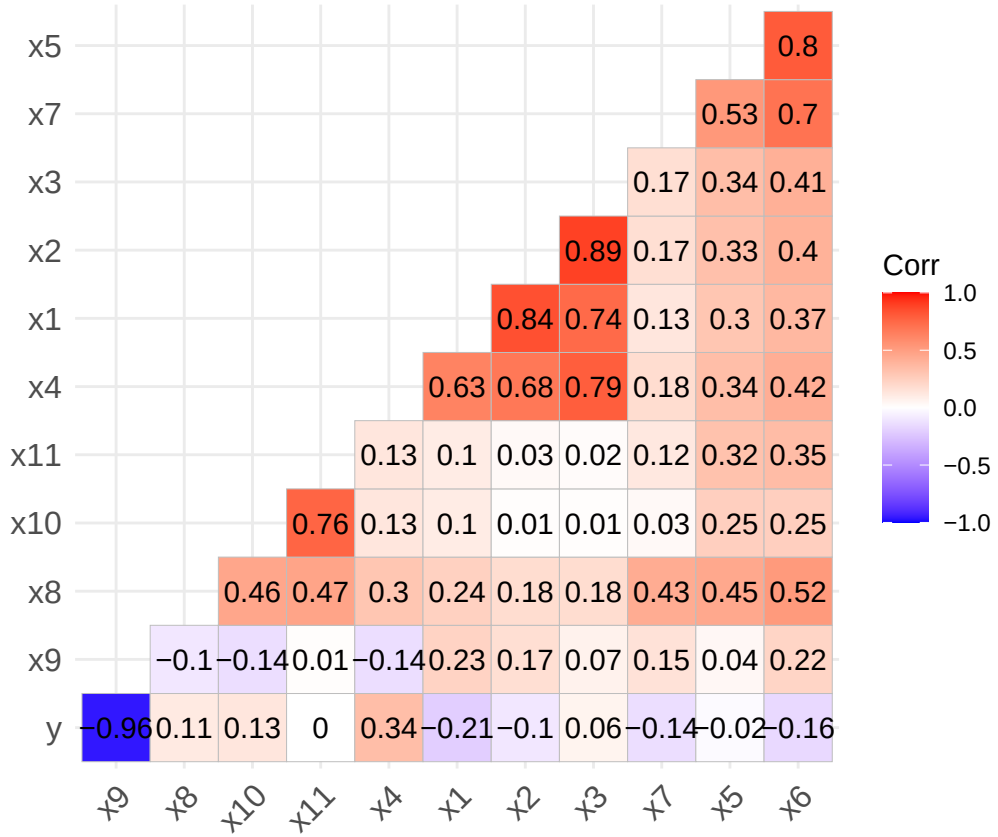
```
##  3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.000   Max.   :1.0000   Max.   :1.0000
##        x5               x6              x7               x8
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.2103   Mean   :0.1436   Mean   :0.1718   Mean   :0.3795
##  3rd Qu.:0.1000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##        x9              x10              x11               y
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :-1.5839
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:-0.8822
##  Median :0.0000   Median :0.0000   Median :0.0000   Median : 1.0154
##  Mean   :0.3821   Mean   :0.2846   Mean   :0.3362   Mean   : 0.3907
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.: 1.4179
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   : 2.0153
```

Seems like the data was loaded correctly.

Before building a linear model, let's create a correlation matrix to verify two things: 1. How each independent variable relates to the output (y) 2. If variables have a high correlation to each other 2.1. If this is the case, we might choose to remove some of the co-correlated variables from the model.

```
correlation_matrix = round(cor(df), 2)

ggcorrplot(correlation_matrix, hc.order = TRUE, type = "lower",
           lab = TRUE)
```



First, looking at the Y row, we can notice that X9 has a very strong negative correlation with the output. Furthermore, X11 has no impact at all; we can assume that X9 will likely be included in the model, while X11 is virtually useless.

Building models

Multiple linear regression

Let us first build a model from the data including all variables.

```
multi_reg <- lm(y ~ x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11, df)
summary(multi_reg)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
##     x10 + x11, data = df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.065561 -0.016967 -0.004286  0.005696  0.137451
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.0425938  0.0163262  63.860  < 2e-16 ***
## x1          -0.6051151  0.0319136 -18.961  < 2e-16 ***
## x2          -0.0006657  0.0461120  -0.014  0.98859
## x3           0.0009480  0.0435062   0.022  0.98278
## x4           0.9839866  0.0297268  33.101  < 2e-16 ***
## x5          -0.0233147  0.0352993  -0.660  0.51454
## x6          -0.1104949  0.0535543  -2.063  0.04883 *
## x7          -0.0955077  0.0323420  -2.953  0.00644 **
## x8           0.0096887  0.0224771   0.431  0.66986
## x9          -2.0191918  0.0204158 -98.903  < 2e-16 ***
## x10          0.0065241  0.0287972   0.227  0.82248
## x11         -0.0212422  0.0275908  -0.770  0.44804
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04911 on 27 degrees of freedom
## Multiple R-squared:  0.9988, Adjusted R-squared:  0.9983
## F-statistic:  2037 on 11 and 27 DF,  p-value: < 2.2e-16
```

With a sample of N $>$ 30, we will not check for the normality of residuals of this model. The R-Squared value of this model is over 0.99, which is a positive indicator but not sufficient to determine this model is good.

The results of this model indicate that the variables X1, X4 and X9 are likely to have a high impact on the output; x6 and x7 possibly as well.

```
multi_reg2 <- lm(y ~ x1+x4+x6+x7+x9, df)
summary(multi_reg2)
```

```
##
## Call:
## lm(formula = y ~ x1 + x4 + x6 + x7 + x9, data = df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.066152 -0.019275 -0.005221  0.004098  0.145236
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  1.03746    0.01251   82.939  < 2e-16 ***
## x1          -0.60424    0.02084  -28.996  < 2e-16 ***
## x4           0.98605    0.02193   44.971  < 2e-16 ***
## x6          -0.14067    0.03412   -4.123 0.000237 ***
## x7          -0.08717    0.02821   -3.090 0.004050 **
## x9          -2.01834    0.01736 -116.282  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04548 on 33 degrees of freedom
## Multiple R-squared:  0.9987, Adjusted R-squared:  0.9985
## F-statistic:  5225 on 5 and 33 DF,  p-value: < 2.2e-16
```

We can see that the adjusted R-Squared value of the second model is slightly higher than the first model's.

Let's use ANOVA to verify whether the added complexity of the first model is significant. From the documentation of the function:

> The anova() function will take the model objects as arguments, and return an ANOVA testing whether the more complex model is significantly better at capturing the data than the simpler model. If the resulting p-value is sufficiently low (usually less than 0.05), we conclude that the more complex model is significantly better than the simpler model, and thus favor the more complex model. If the p-value is not sufficiently low (usually greater than 0.05), we should favor the simpler model.

```
anova(multi_reg, multi_reg2)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11
## Model 2: y ~ x1 + x4 + x6 + x7 + x9
##   Res.Df      RSS Df  Sum of Sq      F Pr(>F)
## 1     27 0.065106
## 2     33 0.068248 -6 -0.0031422 0.2172  0.968
```

Here, ANOVA has tested whether the variables X2, X3, X5, X7, X10 and X11 were relevant. As the p-value is high, we can conclude that that is not the case, so we can stick with the simpler model.

Conclusion

This was a very simplistic approach for tackling the challenge. I was not so familiar with ANOVA, therefore I applied one of its "safe" use cases in this analysis.

A larger sample could have aided this analysis produce more complete results; however, we were able to at least conclude that the unknown system we analyzed can be "fit" without the need of all its inputs.