Tutorial 5 - Experimental Design and Conjoint Analysis

Calum McConville

10/06/2021

5.0 R implementation

Several dependencies will be required to run this workshop. Please install the following packages in R.

```
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
```

```
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(readr)
library(nnet)
library(dplyr)
library(ggplot2)
```

5.1 Fractional Factorial Design

Experiments can be expensive to run.

Taking an example of 4 factors, each with 2 levels, the total number of runs without replicates required to produce a full factorial design - where every combination of factors was explored - would be 2^4 = 16. This can escalate very quickly with factors of greater than 2 levels making it prohibitive to run a full factorial design.

Fractional factorial designs allow you to economically investigate the same cause and effect relationships with fewer samples by taking the assumption that higher order interaction effects are negligible, allowing us to 'alias' our main effects with these higher order terms.

Example
In the following example we will consider the case where, as product analysts at a technology company, we are interested in understanding what features will make our platform more attractive to our users.

We will consider 3 factors, each wil 2 levels: * Team collaboration function (Yes/No) - Called A * Unlimited users per account (Yes/No) - Called B * Unlimited data storage (Yes/No) - Called C

In a full factorial design this would require $2^(3) = 8$ runs to perform a single replicate. A full factorial, orthogonal design with 8 runs can be seen below

```
full_factorial = FrF2(nruns=8, nfactors=3, randomize=F)
```

```
## creating full factorial with 8 runs ...
```

```
print(full_factorial)
```

```
##    A  B  C
## 1 -1 -1 -1
## 2  1 -1 -1
## 3 -1  1 -1
## 4  1  1 -1
## 5 -1 -1  1
## 6  1 -1  1
## 7 -1  1  1
## 8  1  1  1
## class=design, type= full factorial
```

This design will allow us to estimate all effects of A, B and C accounting for interactions.

But suppose that we require two replicates and only have resources available to perform 8 runs. Is there a way can still estimate the effects of A, B and C by perfoming 4 runs per replicate?

This design is referred to as a $2^(3-1)$ half-fraction design. This is the same as a $2^(2)$ full factorial experiment.

To estimate an orthogonal half-fraction design we make the assumption that higher order interactions such as the 2 way interaction between A*B is has a negligible effect and we can therefore replace this interaction term in our two factor full factorial experiment with 3rd remaining factor. This is referred to as "aliasing"

The process described above is implemented via the below code:

```
## creating full factorial with 4 runs ...

## [1] "2-factor full factorial design with interaction term"

##    A  B A_B
## 1 -1 -1   1
## 2  1  1   1
## 3 -1  1  -1
## 4  1 -1  -1
## class=design, type= full factorial

## [1] "1/2 fractional factorial design for 3 factors"

##    A  B  C
## 1 -1 -1  1
## 2  1  1  1
## 3 -1  1 -1
## 4  1 -1 -1
## class=design, type= full factorial
```

We can verify the orthogonality of this design by checking that the sum product of the three vectors is equal to 0.

```
sum(as.numeric(levels(fractional_factorial$A)[fractional_factorial$A])
    * as.numeric(levels(fractional_factorial$B)[fractional_factorial$B])
    * as.numeric(levels(fractional_factorial$C)[fractional_factorial$C]))
```

```
## [1] 0
```

In doing this we have designed an experiment that requires only 4 runs to estimate the main effects of our 3 factors. Note, there is some trade-off with precision of estimates that comes with a fractional factorial design as a result of having fewer observations over which to observe our main effects however this is the price we choose to pay for greater efficiency.

5.2 Conjoint Analysis

Conjoint analysis is a market research method utilised to measure customer preference ("Utility") for various products or services based on their various attributes. Conjoint analysis has several key applications that make it a useful tool in understanding product design and customer preference, namely:

- How to optimise product design by combining optimal attributes?
- How to improve sales revenue by selling the optimal mix of products?
- What is the anticipated market share of new products?

  Choice-based conjoint analysis is a common variation of conjoint analysis that mimics customer buying decisions by allowing them to select/buy a single product profile from a selection of products during a survey.

Example

Using our previous example, we obtain the following data from a user survey where each participant (n=200) is presented with one of 4 product profiles based on our fractional factorial design.

Note: the data has been randomly generated so directional relationships may not be intuitive

```
##   ï..Participant.ID Alternative Choice Collab Users Data
## 1                 1           1      0     No   Yes   No
## 2                 1           2      0    Yes   Yes  Yes
## 3                 1           3      1     No    No  Yes
## 4                 1           4      0    Yes    No   No
## 5                 2           1      1     No   Yes   No
## 6                 2           2      0    Yes   Yes  Yes
## 7                 2           3      0     No    No  Yes
## 8                 2           4      0    Yes    No   No
```

Utility estimation

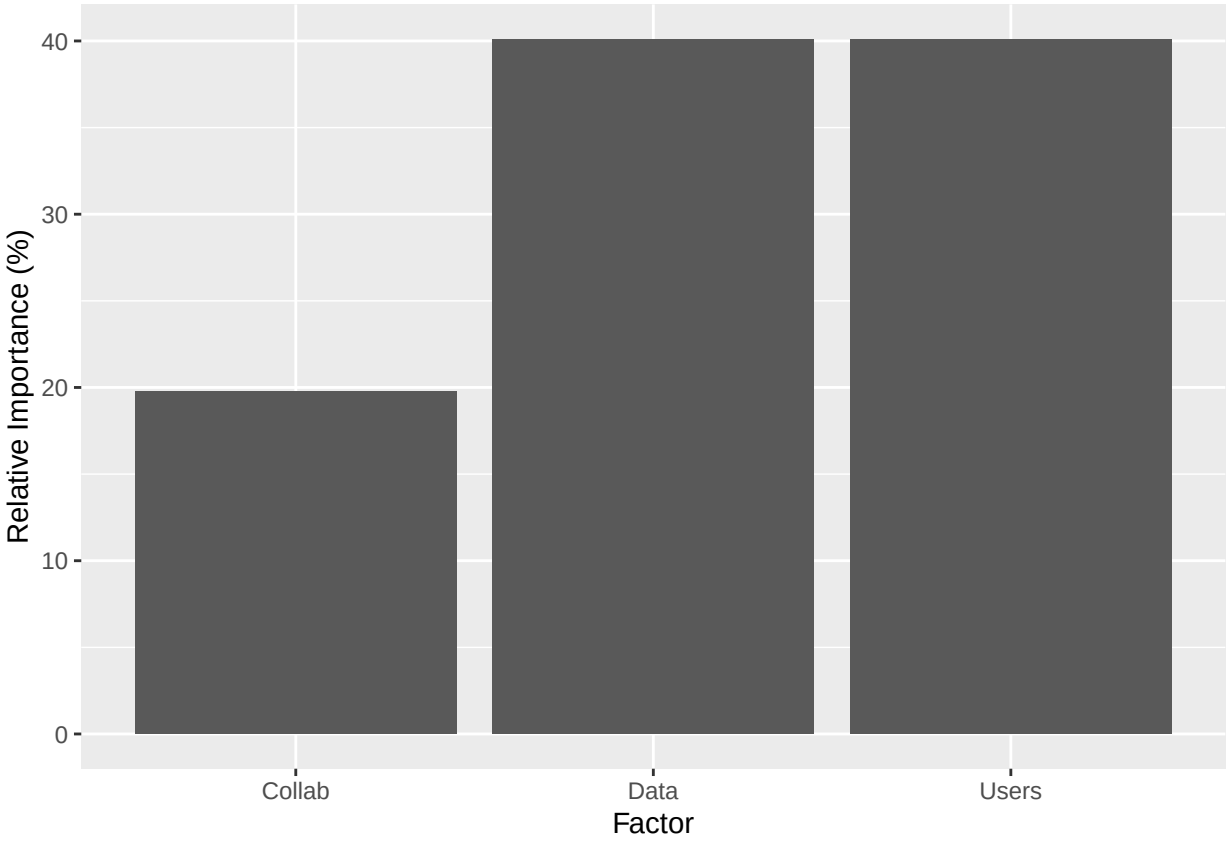
Given we are investigating binary factors, we use a Logit model to calculate the utility of each product attribute with the following form:

$Choice = 0 + \beta_1 * Collab + \beta_2 * Users + \beta_3 * Data$

```
##
## Call:
## glm(formula = Choice ~ Collab + Users + Data, family = binomial(link = "logit"),
##     data = survey_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8782  -0.7320  -0.7320  -0.1376   1.7667
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.8947     0.1649  -5.426 5.75e-08 ***
## CollabYes     0.1409     0.1649   0.855   0.3927
## UsersYes     -0.2856     0.1649  -1.732   0.0833 .
## DataYes      -0.2856     0.1649  -1.732   0.0833 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 899.74  on 799  degrees of freedom
## Residual deviance: 892.53  on 796  degrees of freedom
## AIC: 900.53
```

```
##
## Number of Fisher Scoring iterations: 4
```

The coefficients from our logit model can be interpreted as a customer's utility towards a certain product. The overall importance of each attribute is calculated as the utility range attribute as a proportion of total utility across all attributes.



This shows that both data consumption and user license features are the most important drivers of customer purchase decisions.

Market dyanmics using conjoint analysis

The exponential rule for predicting market share dictates that each option will be chosen with a probability equal to the exponential of its utility over the exponential of total utility across all options e.g. for product 1 $Marketshare = \frac{exp(U_{p1})}{exp(U_{p1})+exp(U_{p2})+exp(U_{p3})+exp(U_{p4})}$.

Remember our four product options were as follows:

```
##   Alternative Collab Users Data
## 1           1     No   Yes   No
## 2           2    Yes   Yes  Yes
## 3           3     No    No  Yes
## 4           4    Yes    No   No
```

Based on the utilities we have calculated previously, the total utility for each of the four products would therefore be:

- $Product1 = 0 + (-0.286) + 0 = -0.286$

- $Product2 = 0.141 + (-0.286) + (-0.286) = -0.431$
- $Product3 = 0 + 0 + (-0.286) = -0.286$
- $Product4 = 0.141 + 0 + 0 = 0.141$

  The market share for each product is therefore:

- $Product1 = \frac{exp(-0.286)}{exp(-0.286)+exp(-0.431)+exp(-0.286)+exp(0.141)} = 22.7\%$
- $Product2 = 19.7\%$
- $Product3 = 22.7\%$
- $Product4 = 34.9\%$

  Extention Question: how would you calculate the market share of these 4 products if a new product was introduced into the market?

Next steps

There are several extensions that have not been captured in this tutorial that will be covered in future lectures:

1. We need to consider what potential unobserved confounding variables may be inherent in our effects due to characteristics of our sample customers e.g. what if our sample population are all paying subscribers to our platform? Is there a common factor shared by these individuals that makes them more likely to preference one response more than another?

- We will explore propensity scoring techniques and instrumental variables in lectures 7 and 8

2. What if we captured our data from historical usage patterns where users were exposed to a different combination of the features described each time? Is there a way of accounting for the available choice set in our analysis?

- We will explore the differences of conjoint analysis on observational vs experimental data in lecture 6