```
---
title: "A.B testing sample project 2"
author: "Aditi Teriar"
date: "10/31/2020"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)


#lets first make up a fake list of  IDS from 1 to 1000 and 1000 random
variables drawn from a normal distribution
df = data.frame("ID"=seq(1,1000), "randomvariable"=rnorm(1000))
#lets now make a function to do the work - you can copy paste this
function into your own scripts
# it needs to be given a dataframe and a list of naming options
# Options might be "treatment" and "control", or if there are more than 2
options then it might be "Control", "treatment1", "treatment2", or just
"LayoutA", "LayoutB"
RCT_random = function(dataframey, values_to_add){

  set.seed(111)
  dataframey$values_to_add[sample(1:nrow(dataframey), nrow(dataframey),
FALSE)] <- rep(values_to_add)
  colnames(dataframey)[which(colnames(dataframey)=="values_to_add")] =
"Status"
  return(dataframey) }
# so this will take the dataframe called "df" and randomly assign each
ROW to "Treatment" or "control"
df_new = RCT_random(df, c("Treatment","Control"))

# make some fake data

#this data will have an ID number, variableA (our variable of interest),
the bank name and the district name
df = data.frame(ID=seq(1,100), variableA=rnorm(100,500,50),
bank_name=c("first_bank","second_bank","third_bank","last_bank"),
District=c("A","B"))

library(knitr)
kable(df[1:5,], format="markdown", align="c")

ICC_CI <- function(cluster_level,outcomevar, dataf){

  #load library
require(ICC)
  set.seed(123)
  si = round(dim(dataf)[1]*0.66)
  values_f <- c()
  for(i in seq(1:50)){
  samp_f = dataf[sample(nrow(dataf), si), ]
  x_f = ICCbare(cluster_level,outcomevar,samp_f)
  values_f <- c(values_f, x_f)
```
```

```
  }
  # note that 1.96StDevs = 95% confidence interval bounds in a normal
dist.
   ret = data.frame("Mean ICC" = round(mean(values_f, na.rm=TRUE),3),
"CI" = round(1.96*sd(values_f, na.rm=TRUE),3))
   ret$Significant = ifelse(ret$Mean.ICC > ret$CI, "Y", "N")
  return( ret)

  }

stored_ICC <- ICC_CI("bank_name", "variableA", df)

ICC_correction <- function(samplesize, num_clusters, ICC_estimate){

  average_cluster_size = samplesize/num_clusters

  factor_inflate = 1 + (average_cluster_size - 1) * ICC_estimate

  return(data.frame("New sample size"=round(samplesize*factor_inflate),
"Old sample size"=samplesize   ))
}

ICC_correction(200, 50, 0.2)

scenario1 <- ICC_correction(200,20,0.2)$New.sample.size #average 10
farmers per cluster, and  20 clusters

scenario2 <- ICC_correction(200,40,0.2)$New.sample.size # average 5
farmers per cluster, and 40 clusters

#make some fake data, let's pretend its baseline data

dat <- data.frame("mean.client.expenditure" = rnorm(1000,100,10))



#this function will plot MDE for various differences
# differs is a list of intervention effects that you want to consider
plot_MDE <- function(historical_data, differs){

  #initialise empty vec
p <- c()


#remember our effect size function from post 1?
cohen_d <- function(d1,d2) {
  m1 <- mean(d1, na.rm=TRUE)
  m2 <- mean(d2, na.rm=TRUE)
  s1 <- sd(d1, na.rm=TRUE)
  s2 <- sd(d2, na.rm=TRUE)
  spo <- sqrt((s1**2 + s2**2)/2)
  d <- (m1 - m2)/spo
  rpb <- d / sqrt((d**2)+4)
  ret <- list("rpb" = rpb, "effectsi" = d)
```

```
  return(ret)  }

#load libs
require(pwr)


for(i in seq(1:length(differs) ) ) {
  samp1 <- historical_data
  xnu = differs[[i]]
  #this is a better version if you can understand it:
    samp2 <- samp1 + rnorm(length(samp1), xnu, xnu/10) #add some noise
  inp <- cohen_d(samp1, samp2)


  p[i] <- pwr.2p.test(h=inp$effectsi , sig.level=0.05, power=0.8,
n=NULL)$n

}

require(ggplot2)
print(ggplot() + geom_point(aes(x=p, y= differs), size=3, color="blue",
shape=1) + geom_line(aes(x=p, y=differs), size=1.2, color="blue") +
xlab("Sample size")+ ylab("MDE") + ggtitle("Minimum detectable effect vs.
sample size"))

library(knitr)
mde_tab = data.frame("MDE"=differs, "Sample size"=p)
kable(mde_tab, digits=2)


}

#set some differences for the loop, here, 1,2,5 (etc) are dollar
increases
diffs <- c(1,2,5,7.5,10,15,20,25)


#plot
plot_MDE(dat$mean.client.expenditure, diffs)


```