

## 0 Preamble

```
#####
# SECTION 0 - PREAMBLE
#####
# load packages
library(dplyr)
library(readr)
library(lfe)
library(lubridate)
library(broom)
library(purrr)
library(ggplot2)

myGGTheme <-
  theme(panel.background = element_rect(fill = NA),
        panel.border = element_rect(fill = NA, color = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.ticks = element_line(color = "gray5"),
        plot.title = element_text(
          face="bold",
          size=19),
        axis.title = element_text(
          color="black",
          face="bold",
          size=15),
        axis.text = element_text(
          color="black",
          size=10)
  )

# set directories
OUT <- file.path("H:", "CoolJoule", "00-Admin", "Design", "PowerCalcs", "Output")
DATA <- file.path("H:", "CoolJoule", "00-Admin", "Design", "PowerCalcs", "Data")

set.seed(4222017)

#####
# END SECTION 0
#####
```

## 1 Prepare Data for Power Calculations

```
#####
# SECTION 1 - PREPARE FOR CALCS
#####
# load data
pilotData <- read.csv(file.path(DATA, "data_Derek.csv"))
pilotData <- pilotData %>%
  select(Casa, Mes, Ano, fecha, treatment, report_intervention_month,
         intervention_group, sms_intervention_month, sms_intervention_month_text,
         Lugar, energia)
```

```
pilotData <- pilotData %>% mutate(sampleMonth = as.factor(paste0(year(fecha), "-", month(fecha))))
```

Diego: Please verify that my definition for the treatment indicator is OK – it is included in the chunk below. I use treatedSMS throughout – which is = 1 in post-treatment period for treatment households only.

```
# create treatment indicators (Check with Diego on this...)
pilotData <- pilotData %>%
  mutate(treatedSMS =
    ifelse(intervention_group == "Treatment Post-Intervention", 1, 0),
  treatedSMS2 =
    ifelse(treatment == "Treatment" &
      sms_intervention_month == 1 &
      Ano > 2015, 1, 0),
  treated = ifelse(treatment == "Treatment", 1, 0))

# balance the data
obsCount <- plyr::count(pilotData$Casa)
pilotDataBalanced <- pilotData %>% filter(Ano > 2014)
obsCount <- plyr::count(pilotDataBalanced$Casa)
pilotDataBalanced <- merge(pilotDataBalanced, obsCount, by.x = "Casa", by.y = "x")
pilotDataBlaanced <- pilotDataBalanced %>% filter(freq == 25)
#####
# END SECTION 1
#####
```

## 2 Conduct Preliminary Analysis on Pilot Data

```
#####
# SECTION 2 - GET TREATMENT EFFECTS
#####
# 2.1 -- treatment effect for SMS treatment in unbalanced sample
smsEffectLevel <- feIm(energia ~ treatedSMS | Casa + sampleMonth | 0 | Casa, data = pilotData)
summary(smsEffectLevel)

##
## Call:
## feIm(formula = energia ~ treatedSMS | Casa + sampleMonth | 0 | Casa, data = pilotData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -254.801  -18.499   2.094   18.786  140.761
##
## Coefficients:
##              Estimate Cluster s.e. t value Pr(>|t|)
## treatedSMS    -14.00         17.27  -0.811    0.418
##
## Residual standard error: 42.03 on 713 degrees of freedom
## Multiple R-squared(full model): 0.8379   Adjusted R-squared: 0.8245
## Multiple R-squared(proj model): 0.006028   Adjusted R-squared: -0.07622
## F-statistic(full model, *iid*):62.48 on 59 and 713 DF, p-value: < 2.2e-16
## F-statistic(proj model): 0.6574 on 1 and 29 DF, p-value: 0.4241

smsLevelEffect <- smsEffectLevel$beta[1]
## get percent change
```

```

ctrlMean <- pilotData %>% group_by(intervention_group) %>%
  summarise(mean(energia))
percentEffectPreTreatment <- smsLevelEffect/ctrlMean$`mean(energia)`[2]
percentEffectPostTreatment <- smsLevelEffect/ctrlMean$`mean(energia)`[1]

## this is MDE
percentEffectPreTreatment # percent using pre-treatment control as comparison
## [1] -0.06817607

percentEffectPostTreatment # percent using post-treatment control as comparison
## [1] -0.06377832

## now use logs (which approximately percent change and controls for outliers)
smsEffectLog <- felm(log(energia) ~ treatedSMS | Casa + sampleMonth | 0 | Casa, data = pilotData)
summary(smsEffectLog)

##
## Call:
##   felm(formula = log(energia) ~ treatedSMS | Casa + sampleMonth |      0 | Casa, data = pilotData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.45588 -0.08746  0.01010  0.11454  0.68425
##
## Coefficients:
##              Estimate Cluster s.e. t value Pr(>|t|)
## treatedSMS -0.03367      0.09597  -0.351    0.726
##
## Residual standard error: 0.2683 on 713 degrees of freedom
## Multiple R-squared(full model): 0.7464   Adjusted R-squared: 0.7254
## Multiple R-squared(proj model): 0.0008601   Adjusted R-squared: -0.08182
## F-statistic(full model, *iid*):35.57 on 59 and 713 DF, p-value: < 2.2e-16
## F-statistic(proj model): 0.1231 on 1 and 29 DF, p-value: 0.7283

## this is MDE in percent from log equation
percentEffectLog <- smsEffectLog$beta[1]
percentEffectLog
## [1] -0.03366871

# 2.2 -- treatment effect for SMS treatment in balanced sample
smsEffectLevelBalanced <- felm(energia ~ treatedSMS | Casa + sampleMonth | 0 | Casa, data = pilotDataBalanced)
summary(smsEffectLevelBalanced)

##
## Call:
##   felm(formula = energia ~ treatedSMS | Casa + sampleMonth | 0 | Casa, data = pilotDataBalanced)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -254.722  -17.518    2.146   18.275  140.593
##
## Coefficients:
##              Estimate Cluster s.e. t value Pr(>|t|)
## treatedSMS  -14.45      17.05  -0.848    0.397
##

```

```
## Residual standard error: 40.08 on 670 degrees of freedom
## Multiple R-squared(full model): 0.8501    Adjusted R-squared: 0.838
## Multiple R-squared(proj model): 0.007461    Adjusted R-squared: -0.07253
## F-statistic(full model, *iid*):70.37 on 54 and 670 DF, p-value: < 2.2e-16
## F-statistic(proj model): 0.7185 on 1 and 29 DF, p-value: 0.4036

smsLevelEffectBalanced <- smsEffectLevelBalanced$beta[1]

## get percent change
ctrlMean <- pilotDataBalanced %>% group_by(intervention_group) %>%
  summarise(mean(energia))
percentEffectPreTreatmentBalanced <- smsLevelEffectBalanced/ctrlMean$`mean(energia)`[2]
percentEffectPostTreatmentBalanced <- smsLevelEffectBalanced/ctrlMean$`mean(energia)`[1]

## these are MDEs based on
percentEffectPreTreatmentBalanced # percent using pre-treatment control as comparison
## [1] -0.06981994

percentEffectPostTreatmentBalanced # percent using post-treatment control as comparison
## [1] -0.06584819

## do it in logs too as before.
smsEffectLogBalanced <- feIm(log(energia) ~ treatedSMS | Casa + sampleMonth | 0 | Casa, data = pilotDataBalanced)
summary(smsEffectLogBalanced)

##
## Call:
## feIm(formula = log(energia) ~ treatedSMS | Casa + sampleMonth | 0 | Casa, data = pilotDataBalanced)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.60738 -0.08049  0.01255  0.11213  0.65140
##
## Coefficients:
##              Estimate Cluster s.e. t value Pr(>|t|)
## treatedSMS -0.03829      0.09344  -0.41    0.682
##
## Residual standard error: 0.2479 on 670 degrees of freedom
## Multiple R-squared(full model): 0.7699    Adjusted R-squared: 0.7514
## Multiple R-squared(proj model): 0.001377    Adjusted R-squared: -0.07911
## F-statistic(full model, *iid*):41.52 on 54 and 670 DF, p-value: < 2.2e-16
## F-statistic(proj model): 0.1679 on 1 and 29 DF, p-value: 0.685

## this is MDE
percentEffectLogBalanced <- smsEffectLogBalanced$beta[1]

#notes as of 4/24/2017:
# MDE - UNBALANCED SAMPLE
## PRE-TREATMENT CONTROL GROUP AS COMPARISON
percentEffectPreTreatment
## [1] -0.06817607

## POST-TREATMENT CONTROL GROUP AS COMPARISON
percentEffectPostTreatment
```

```
## [1] -0.06377832
## LOG SPECIFICATION
percentEffectLog
## [1] -0.03366871
# MDE - BALANCED SAMPLE
## PRE-TREATMENT CONTROL GROUP AS COMPARISON
percentEffectPreTreatmentBalanced
## [1] -0.06981994
## POST-TREATMENT CONTROL GROUP AS COMPARISON
percentEffectPostTreatmentBalanced
## [1] -0.06584819
## LOG SPECIFICATION
percentEffectLogBalanced
## [1] -0.03828541
## overall range
range(c(percentEffectPreTreatment, percentEffectPostTreatment,smsEffectLog$beta[1],
        percentEffectPreTreatmentBalanced,percentEffectPostTreatmentBalanced,smsEffectLogBalanced$beta[1]))
## [1] -0.06981994 -0.03366871
#####
# END SECTION 2
#####
```

### 3 Run Power Calculations - Simulation #1

This procedure does the following:

1. Set  $N = N_0$
2. Sample (with replacement)  $N$  households from the treatment group and  $N$  households from the control group (total sample size =  $2N$ ).
3. Run a regression of energy on the treatment indicator and Casa and sampleMonth fixed effects (preferred specification is in logs with Casa clustering)
4. Repeat (2)-(3) 500 times and count the number of times the treatment effect is significant at  $p = 0.05$
5. Record percent of significant coefficients
6. Increase  $N$  by 25
7. If percent  $\geq 80\%$ , run steps (2)-(5) 20 more times and then conclude.
8. If percent is  $\leq 80\%$  restart at step (2).

The percent of significant coefficients is the "power" of the test. The plots show power versus sample size. The preferred specification is in logs with household level clustering.

```
#####
# SECTION 3 - POWER CALCS SIM 1
#####
# SIMULATE POWER CALCS BY SAMPLING
```

```

# WITH REPLACEMENT N HOUSEHOLDS
# FROM BOTH TREATMENT/CONTROL GROUP
# RUN K REGRESSIONS AND COUNT HOW
# MANY TIMES WE GET A SIGNIF. P-VALUE

# Create sampling function with regression inner loop
nFinder <- function(N, cluster, logs, levels){
  # grab list of hhid ids
  controlHHs <- pilotDataBalanced %>%
    filter(treatment == "Control") %>% distinct(Casa)
  treatmentHHs <- pilotDataBalanced %>%
    filter(treatment == "Treatment") %>% distinct(Casa)
  # sample N observations with replacement
  bootSampleControl <- sample_n(controlHHs, size = N, replace = TRUE)
  bootSampleTreatment <- sample_n(treatmentHHs, size = N, replace = TRUE)
  bootSample <- rbind(bootSampleControl, bootSampleTreatment)
  bootSample <- bootSample %>% mutate(bootID = row_number())

  # create dataset based on sampled casas
  # Create dataset based on sampled hhids +
  bootData <- filter(pilotDataBalanced, Casa %in% bootSample$Casa) %>%
    merge(bootSample, by = "Casa")

  # run a regression of kwh on treatment indicator to test for significance of
  # estimated treated effect coefficient
  if(cluster == FALSE & levels == TRUE){
    nReg <- felm(energia ~ treatedSMS | bootID + sampleMonth | 0 | 0, data = bootData)
  }
  if(cluster == FALSE & logs == TRUE){
    nReg <- felm(log(energia) ~ treatedSMS | bootID + sampleMonth | 0 | 0, data = bootData)
  }
  if(cluster == TRUE & levels == TRUE){
    nReg <- felm(energia ~ treatedSMS | bootID + sampleMonth | 0 | bootID, data = bootData)
  }
  if(cluster == TRUE & logs == TRUE){
    nReg <- felm(log(energia) ~ treatedSMS | bootID + sampleMonth | 0 | bootID, data = bootData)
  }
  # boolean to check if p-value of treatment effect coef. est. is <0.05
  tidy(nReg)$p.value < .05 %>%
    return()
}

powerCalcSim <- function(nOrig, stepSize, simIterations,
                        clusterSwitch, levelSwitch, logSwitch){
  # create empty powerDB
  powerDB <- data.frame(sampleSize = integer(), power = double())

  # set N to nOrig
  N <- nOrig

  # loop over power until >=.8
  power <- 0
  while(power < 0.8){

```

```

#print(N)
power <- mean(map_lgl(rep(N,simIterations),nFinder,
                      cluster = clusterSwitch, levels = levelSwitch, logs = logSwitch))

#print(power)
temp <- data.frame(sampleSize = N*2, Power = power)
powerDB <- rbind(powerDB, temp)
N <- N + stepSize
}
for(i in 1:20){
#print(N)
power <- mean(map_lgl(rep(N,simIterations),nFinder,
                      cluster = clusterSwitch, levels = levelSwitch, logs = logSwitch))

#print(power)
temp <- data.frame(sampleSize = N*2, Power = power)
powerDB <- rbind(powerDB, temp)
N <- N + stepSize
}
return(powerDB)
}

```

```

n0 <- 25
step <- 25
iterations <- 500

pwr1 <- powerCalcSim(n0,step,iterations,
                    clusterSwitch = FALSE, levelSwitch = TRUE, logSwitch = FALSE)
pwr2 <- powerCalcSim(n0,step,iterations,
                    clusterSwitch = TRUE, levelSwitch = TRUE, logSwitch = FALSE)
pwr3 <- powerCalcSim(n0,step,iterations,
                    clusterSwitch = FALSE, levelSwitch = FALSE, logSwitch = TRUE)
pwr4 <- powerCalcSim(n0,step,iterations,
                    clusterSwitch = TRUE, levelSwitch = FALSE, logSwitch = TRUE)

```

```

# export the results
write_excel_csv(pwr1, file.path(OUT,"SET1-noClusterLevels.csv"))
write_excel_csv(pwr2, file.path(OUT,"SET1-clusterLevels.csv"))
write_excel_csv(pwr3, file.path(OUT,"SET1-noClusterLogs.csv"))
write_excel_csv(pwr4, file.path(OUT,"SET1-clusterLogs.csv"))

```

```

#plot some results
plot1 <- ggplot(data = pwr1, aes(sampleSize, Power)) +
  geom_smooth() + geom_point() + geom_hline(yintercept=.8) +
  myGGTheme +
  ggtitle("No Clustering - Levels") +
  xlab("Sample Size") +
  scale_x_continuous(breaks = seq(0,range(pwr1$sampleSize)[2],100)) +
  ylab("Power") +
  scale_y_continuous(breaks = seq(0,1,.1))

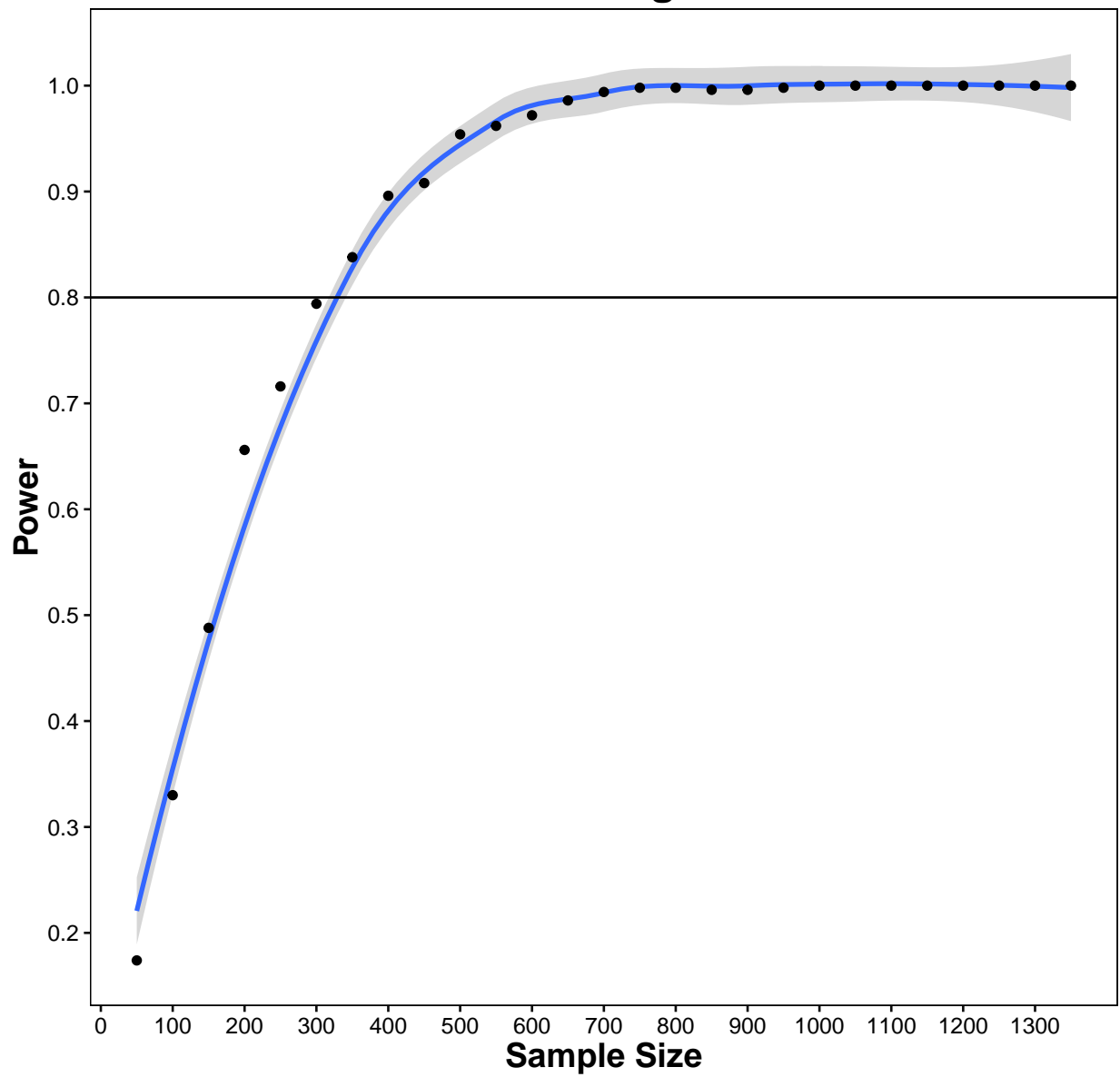
plot2 <- ggplot(data = pwr2, aes(sampleSize, Power)) +
  geom_smooth() + geom_point() + geom_hline(yintercept=.8) +
  myGGTheme +
  ggtitle("Casa Clustering - Levels") +
  xlab("Sample Size") +

```

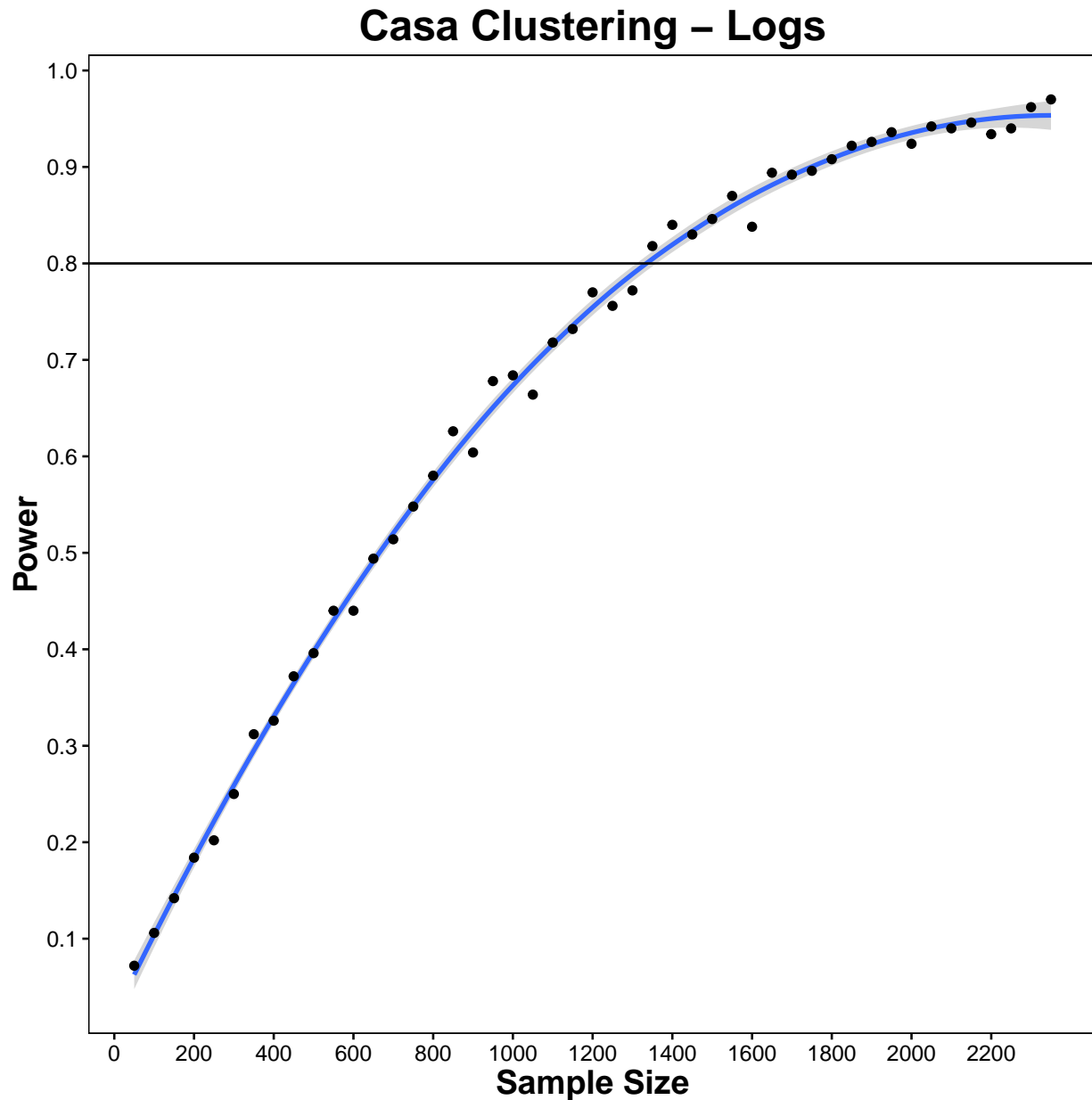
```
scale_x_continuous(breaks = seq(0,range(pwr2$sampleSize)[2],100)) +  
ylab("Power") +  
scale_y_continuous(breaks = seq(0,1,.1))  
  
plot3 <- ggplot(data = pwr3, aes(sampleSize, Power)) +  
  geom_smooth() + geom_point() + geom_hline(yintercept=.8) +  
  myGGTheme +  
  ggtitle("No Clustering - Logs") +  
  xlab("Sample Size") +  
  scale_x_continuous(breaks = seq(0,range(pwr3$sampleSize)[2],200)) +  
  ylab("Power") +  
  scale_y_continuous(breaks = seq(0,1,.1))  
  
plot4 <- ggplot(data = pwr4, aes(sampleSize, Power)) +  
  geom_smooth() + geom_point() + geom_hline(yintercept=.8) +  
  myGGTheme +  
  ggtitle("Casa Clustering - Logs") +  
  xlab("Sample Size") +  
  scale_x_continuous(breaks = seq(0,range(pwr4$sampleSize)[2],200)) +  
  ylab("Power") +  
  scale_y_continuous(breaks = seq(0,1,.1))  
  
plot2
```



## Casa Clustering – Levels



plot4



## 4 Run Power Calculations - Simulation #2

This procedure does the following:

1. Set  $N = N_0$
2. Sample (with replacement)  $N$  households from the control group
3. Randomly assign half of the households to the treatment group and apply the relevant treatment effect (as estimated from the preliminary data) to the treated household
4. Run a regression of energy on the treatment indicator and Casa and sampleMonth fixed effects (preferred specification is in logs with Casa clustering)

5. Repeat (2)-(3) 500 times and count the number of times the treatment effect is significant at  $p = 0.05$
6. Record percent of significant coefficients
7. Increase  $N$  by 50
8. If percent  $\geq 80\%$ , run steps (2)-(5) 20 more times and then conclude.
9. If percent is  $\leq 80\%$  restart at step (2).

The percent of significant coefficients is the "power" of the test. The plots show power versus sample size. The preferred specification is in logs with household level clustering.

```
# now run them with a fixed MDE and sample the control households
# Create sampling function with regression inner loop
nFinder2 <- function(N, MDE, cluster, logs, levels){
  # grab list of control hhid ids
  controlHHs <- pilotDataBalanced %>%
    filter(treatment == "Control") %>% distinct(Casa)

  # sample N observations from control HHs with replacement
  bootSample <- sample_n(controlHHs, size = N, replace = TRUE) %>%
    mutate(bootID = row_number(),
           inTreatment = sample(0:1, N, replace = TRUE))

  # create dataset based on sampled casasa
  bootData <- filter(pilotDataBalanced, Casa %in% bootSample$Casa) %>%
    merge(bootSample, by = "Casa") %>%
    mutate(
      treatedSMS =
        as.numeric(inTreatment == 1 & intervention_group == "Control Post-Intervention"))

  # run a regression of kwh on treatment indicator to test for significance of
  # estimated treated effect coefficient
  if(cluster == FALSE & levels == TRUE){
    bootData <- bootData %>%
      mutate(energia = energia + MDE*treatedSMS)
    nReg <- feelm(energia ~ treatedSMS | bootID + sampleMonth | 0 | 0, data = bootData)
  }
  if(cluster == FALSE & logs == TRUE){
    bootData <- bootData %>%
      mutate(logEnergia = log(energia) + MDE*treatedSMS)
    nReg <- feelm(logEnergia ~ treatedSMS | bootID + sampleMonth | 0 | 0, data = bootData)
  }
  if(cluster == TRUE & levels == TRUE){
    bootData <- bootData %>%
      mutate(energia = energia + MDE*treatedSMS)
    nReg <- feelm(energia ~ treatedSMS | bootID + sampleMonth | 0 | bootID, data = bootData)
  }
  if(cluster == TRUE & logs == TRUE){
    bootData <- bootData %>%
      mutate(logEnergia = log(energia) + MDE*treatedSMS)
    nReg <- feelm(logEnergia ~ treatedSMS | bootID + sampleMonth | 0 | bootID, data = bootData)
  }

  # boolean to check if p-value of treatment effect coef. est. is <0.05
  tidy(nReg)$p.value < .05 %>%
    return()
}
```

```

}

powerCalcSim2 <- function(nOrig, stepSize, simIterations, MDESet,
                          clusterSwitch, levelSwitch, logSwitch){
  # create empty powerDB
  powerDB <- data.frame(sampleSize = integer(), power = double())

  # set N to nOrig
  N <- nOrig

  # loop over power until >=.8
  power <- 0
  while(power < 0.8){
    #print(N)
    power <- mean(map_lgl(rep(N,simIterations),nFinder2, MDE = MDESet,
                          cluster = clusterSwitch, levels = levelSwitch, logs = logSwitch))
    #print(power)
    temp <- data.frame(sampleSize = N, Power = power)
    powerDB <- rbind(powerDB, temp)
    N <- N + stepSize
  }
  for(i in 1:20){
    #print(N)
    power <- mean(map_lgl(rep(N,simIterations),nFinder2, MDE = MDESet,
                          cluster = clusterSwitch, levels = levelSwitch, logs = logSwitch))
    #print(power)
    temp <- data.frame(sampleSize = N, Power = power)
    powerDB <- rbind(powerDB, temp)
    N <- N + stepSize
  }
  return(powerDB)
}

n0 <- 50
step <- 50
iterations <- 500

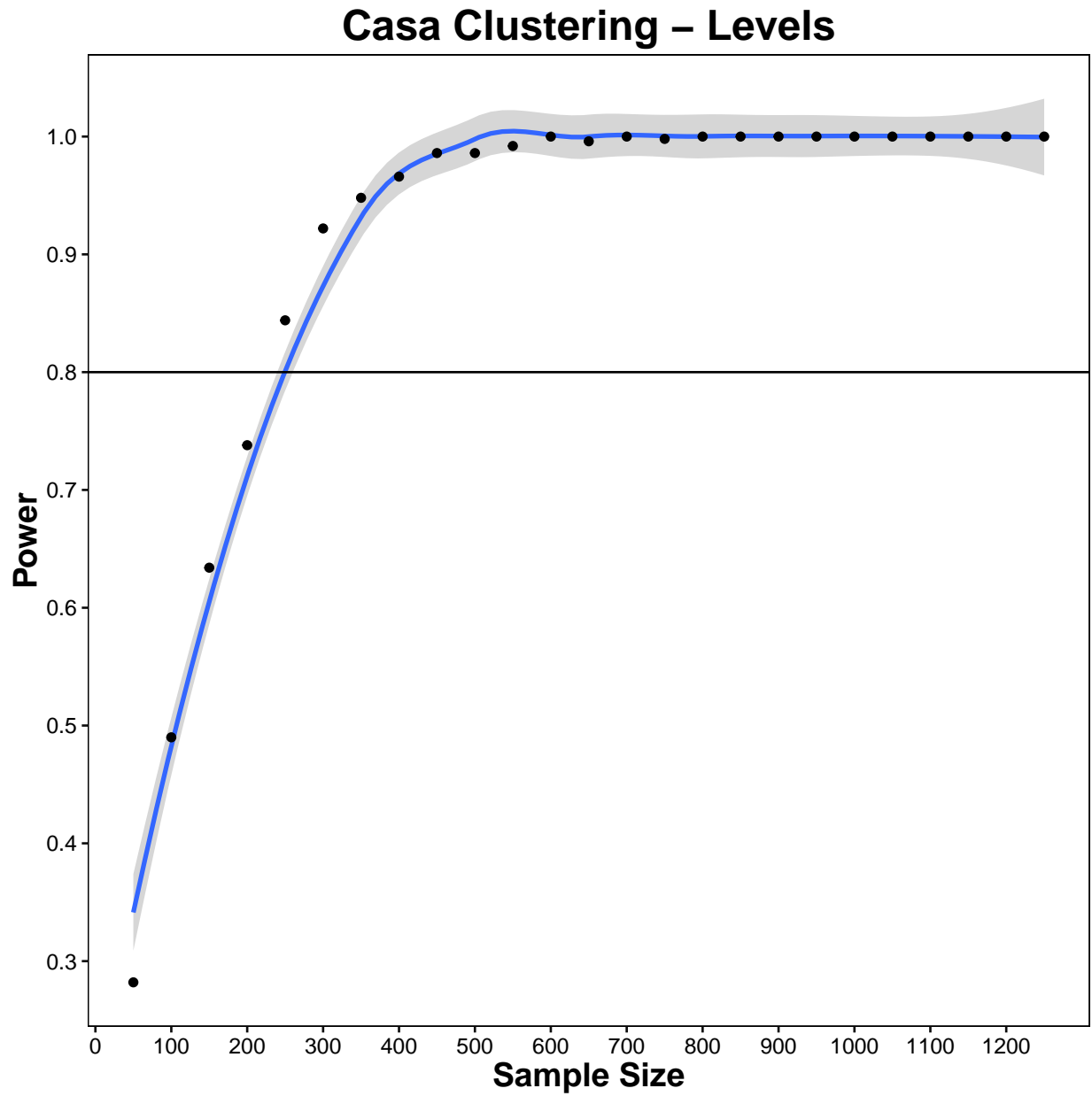
pwr5 <- powerCalcSim2(n0,step,iterations, smsLevelEffectBalanced,
                      clusterSwitch = FALSE, levelSwitch = TRUE, logSwitch = FALSE)
pwr6 <- powerCalcSim2(n0,step,iterations, smsLevelEffectBalanced,
                      clusterSwitch = TRUE, levelSwitch = TRUE, logSwitch = FALSE)
pwr7 <- powerCalcSim2(n0,step,iterations, percentEffectLogBalanced,
                      clusterSwitch = FALSE, levelSwitch = FALSE, logSwitch = TRUE)
pwr8 <- powerCalcSim2(n0,step,iterations,percentEffectLogBalanced,
                      clusterSwitch = TRUE, levelSwitch = FALSE, logSwitch = TRUE)

# export the results
write_excel_csv(pwr5, file.path(OUT,"SET2-noClusterLevels.csv"))
write_excel_csv(pwr6, file.path(OUT,"SET2-clusterLevels.csv"))
write_excel_csv(pwr7, file.path(OUT,"SET2-noClusterLogs.csv"))
write_excel_csv(pwr8, file.path(OUT,"SET2-clusterLogs.csv"))

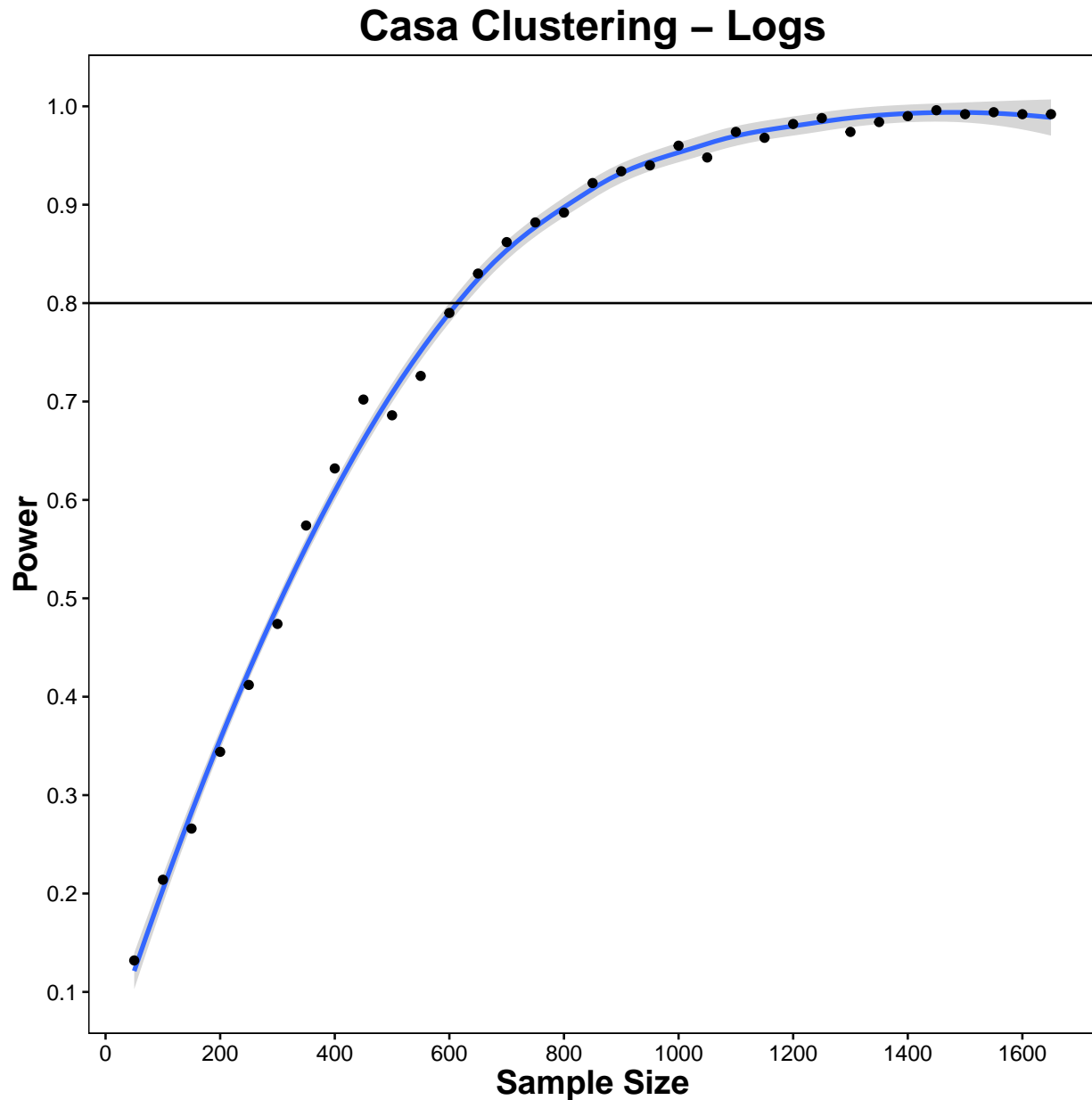
#plot some results
plot5 <- ggplot(data = pwr5, aes(sampleSize, Power)) +
  geom_smooth() + geom_point() + geom_hline(yintercept=.8) +

```

```
myGGTheme +  
  ggtitle("No Clustering - Levels") +  
  xlab("Sample Size") +  
  scale_x_continuous(breaks = seq(0,range(pwr5$sampleSize)[2],100)) +  
  ylab("Power") +  
  scale_y_continuous(breaks = seq(0,1,.1))  
  
plot6 <- ggplot(data = pwr6, aes(sampleSize, Power)) +  
  geom_smooth() + geom_point() + geom_hline(yintercept=.8) +  
  myGGTheme +  
  ggtitle("Casa Clustering - Levels") +  
  xlab("Sample Size") +  
  scale_x_continuous(breaks = seq(0,range(pwr6$sampleSize)[2],100)) +  
  ylab("Power") +  
  scale_y_continuous(breaks = seq(0,1,.1))  
  
plot7 <- ggplot(data = pwr7, aes(sampleSize, Power)) +  
  geom_smooth() + geom_point() + geom_hline(yintercept=.8) +  
  myGGTheme +  
  ggtitle("No Clustering - Logs") +  
  xlab("Sample Size") +  
  scale_x_continuous(breaks = seq(0,range(pwr7$sampleSize)[2],200)) +  
  ylab("Power") +  
  scale_y_continuous(breaks = seq(0,1,.1))  
  
plot8 <- ggplot(data = pwr8, aes(sampleSize, Power)) +  
  geom_smooth() + geom_point() + geom_hline(yintercept=.8) +  
  myGGTheme +  
  ggtitle("Casa Clustering - Logs") +  
  xlab("Sample Size") +  
  scale_x_continuous(breaks = seq(0,range(pwr8$sampleSize)[2],200)) +  
  ylab("Power") +  
  scale_y_continuous(breaks = seq(0,1,.1))  
  
plot6
```



plot8



## 5 Final Thoughts

First and foremost, these calculations assume 100% take-up – i.e. anyone we assign to treatment takes up the treatment. If we do this on a set of households that are interested in adopting the technology – I think we'll be pretty close to 100% take-up. Let's definitely discuss the trade-offs in terms of design/sample size requirements when we meet next.

Now, if only 33% of people offered the treatment take-up the treatment the number of households required increases by a factor of  $\frac{1}{.33}^2 = 9$ . I can go into the details of this calculation – but in essence, if we do some sort of encouragement design and it is not very effective the sample size can increase very rapidly as the percent of take-up decreases.

I'll just discuss the two preferred specification [logs and household clustering of errors] – it seems that using

simulation #1, where I sample treatment households from the treatment sample and control households from the control sample, the required number of household to have 80% power to measure an effect on energy use we'll need about 1400 households (split 50/50 in control). Now in simulation #2, where I sample only from control households and then apply the estimated treatment effect from the pilot data to half of these households, this number drops to about 700. The reason for this discrepancy is that the second simulation allows for no heterogeneity in the effect of the treatment, whereas the first simulation allows for heterogeneity (since we are sampling with replacement, treatment households from those treated in the pilot).

Now, power calculations are hardly a science, so I don't think we need to abide by either of these numbers persay – but I'd argue that if we could have as sample size in the final RCT of around 1200 to 1400 households that would be ideal for measuring energy effects. These numbers would certainly be suitable (and probably overkill) for measuring the non-energy related effects, which I also think would be interesting. However, it's probably best to prepare for the best-case scenario – which would be to go for 1200-1400 households.

Again – these are just a jumping off point and if my definition of the treatment variable is not correct, let me know so I can re-run these calculations. However, these do go to show that we need a fairly large number of households in an RCT to get a nice measure of energy savings from the appliance you guys are manufacturing.

Let's schedule a time to discuss this moving forward. I apologize for not making this document more clear – but I think some face-to-face time with this in hand will help us tremendously (a phone call would work too).