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 Derek
OUT  <- file.path("G:","flexbox-data-dump-analysis","PowerCalcs","Output")
DATA <- file.path("G:","flexbox-data-dump-analysis","PowerCalcs","Data")

# set directories Diego
#OUT  <- file.path("/Users/diego/Desktop/Projects_Code/","flexbox-data-dump-analysis","PowerCalcs","Out
#DATA <- file.path("/Users/diego/Desktop/Projects_Code/","flexbox-data-dump-analysis","PowerCalcs","Dat

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))

# Diego: I think this looks Ok. The only thing I would remove would be January 2017, as we removed all
# making sure that we're only using 2015 and 2016 as we don't know anything about the houses prior to 2
# DereK: Removing 2017 data
pilotData <- pilotData %>% filter(Ano > 2014 & Ano < 2017)

# DEREK: Doesn't look like this code does anything... but keeping it here since
# Diego added on 4/30/17
pilotData <- subset(pilotData,
                    pilotData$Casa!= "A26" |
                    pilotData$Casa!= "A11" |
                    pilotData$Casa!= "A29" |
                    pilotData$Casa!= "45" |
                    pilotData$Casa!= "191" )
```

```
# balance the data
obsCount <- plyr::count(pilotData$Casa)
pilotDataBalanced <- merge(pilotData, obsCount, by.x = "Casa", by.y = "x") %>%
  filter(freq == 24)
###################################
# 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 <- felm(energia ~ treatedSMS | Casa + sampleMonth| 0 | Casa, data = pilotData)
summary(smsEffectLevel)

##
## Call:
##    felm(formula = energia ~ treatedSMS | Casa + sampleMonth | 0 |      Casa, data = pilotData)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -252.209  -16.682    2.697   18.295  141.576
##
## Coefficients:
##            Estimate Cluster s.e. t value Pr(>|t|)
## treatedSMS   -16.75        16.89  -0.991    0.322
##
## Residual standard error: 39.83 on 642 degrees of freedom
## Multiple R-squared(full model): 0.8525   Adjusted R-squared: 0.8404
## Multiple R-squared(proj model): 0.009633   Adjusted R-squared: -0.07213
## F-statistic(full model, *iid*):70.04 on 53 and 642 DF, p-value: < 2.2e-16
## F-statistic(proj model): 0.9829 on 1 and 29 DF, p-value: 0.3297
```

```
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]
#Diego: Don't really get this one. Why assign treatment the mean of the control group?
#Derek 5/1: Getting the percent change in energy use w.r.t. the control group.  I do this since this th


## this is MDE
percentEffectPreTreatment # percent change relative to pre-treatment control as comparison
```

```
## [1] -0.08090233
```

```
percentEffectPostTreatment # percent change relative to post-treatment control as comparison
```

```
## [1] -0.07540445
```

```
## 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
## -1.58912 -0.07734  0.01283  0.11192  0.62948
##
## Coefficients:
##            Estimate Cluster s.e. t value Pr(>|t|)
## treatedSMS -0.04870      0.09341  -0.521    0.602
##
## Residual standard error: 0.2489 on 642 degrees of freedom
## Multiple R-squared(full model): 0.769   Adjusted R-squared: 0.7499
## Multiple R-squared(proj model): 0.002102   Adjusted R-squared: -0.08028
## F-statistic(full model, *iid*):40.32 on 53 and 642 DF, p-value: < 2.2e-16
## F-statistic(proj model): 0.2718 on 1 and 29 DF, p-value: 0.6061
```

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

```
## [1] -0.04869675
```

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

```
##
## Call:
##    felm(formula = energia ~ treatedSMS | Casa + sampleMonth | 0 |      Casa, data = pilotDataBalance
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -254.003  -17.248    3.184   18.948  125.338
##
## Coefficients:
##            Estimate Cluster s.e. t value Pr(>|t|)
## treatedSMS   -13.83        17.63  -0.785    0.433
##
## Residual standard error: 39 on 597 degrees of freedom
## Multiple R-squared(full model): 0.8632   Adjusted R-squared: 0.8518
## Multiple R-squared(proj model): 0.006759   Adjusted R-squared: -0.07643
## F-statistic(full model, *iid*):75.36 on 50 and 597 DF, p-value: < 2.2e-16
## F-statistic(proj model): 0.6155 on 1 and 26 DF, p-value: 0.4398
```

```
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] -0.06660483
```

```
percentEffectPostTreatmentBalanced # percent using post-treatment control as comparison
```

```
## [1] -0.06272937
```

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

```
##
## Call:
##    felm(formula = log(energia) ~ treatedSMS | Casa + sampleMonth |      0 | Casa, data = pilotDataBa
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.60079 -0.08156  0.01456  0.11570  0.63804
##
## Coefficients:
```

```
##            Estimate Cluster s.e. t value Pr(>|t|)
## treatedSMS -0.03461      0.09737  -0.355    0.722
##
## Residual standard error: 0.2516 on 597 degrees of freedom
## Multiple R-squared(full model): 0.7735   Adjusted R-squared: 0.7545
## Multiple R-squared(proj model): 0.001023   Adjusted R-squared: -0.08264
## F-statistic(full model, *iid*):40.78 on 50 and 597 DF, p-value: < 2.2e-16
## F-statistic(proj model): 0.1264 on 1 and 26 DF, p-value: 0.7251
```

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

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

```
## [1] -0.08090233
```

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

```
## [1] -0.07540445
```

```
## LOG SPECIFICATION
percentEffectLog
```

```
## [1] -0.04869675
```

```
# MDE - BALANCED SAMPLE
## PRE-TREATMENT CONTROL GROUP AS COMPARISON
percentEffectPreTreatmentBalanced
```

```
## [1] -0.06660483
```

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

```
## [1] -0.06272937
```

```
## LOG SPECIFICATION
percentEffectLogBalanced
```

```
## [1] -0.03461387
```

```
## overall range
range(c(percentEffectPreTreatment, percentEffectPostTreatment,smsEffectLog$beta[1],
        percentEffectPreTreatmentBalanced,percentEffectPostTreatmentBalanced,smsEffectLogBalanced$beta[
```

```
## [1] -0.08090233 -0.03461387
```

```
###################################
# 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 groupo (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 perferred 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 SIGNF. 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()
}
```

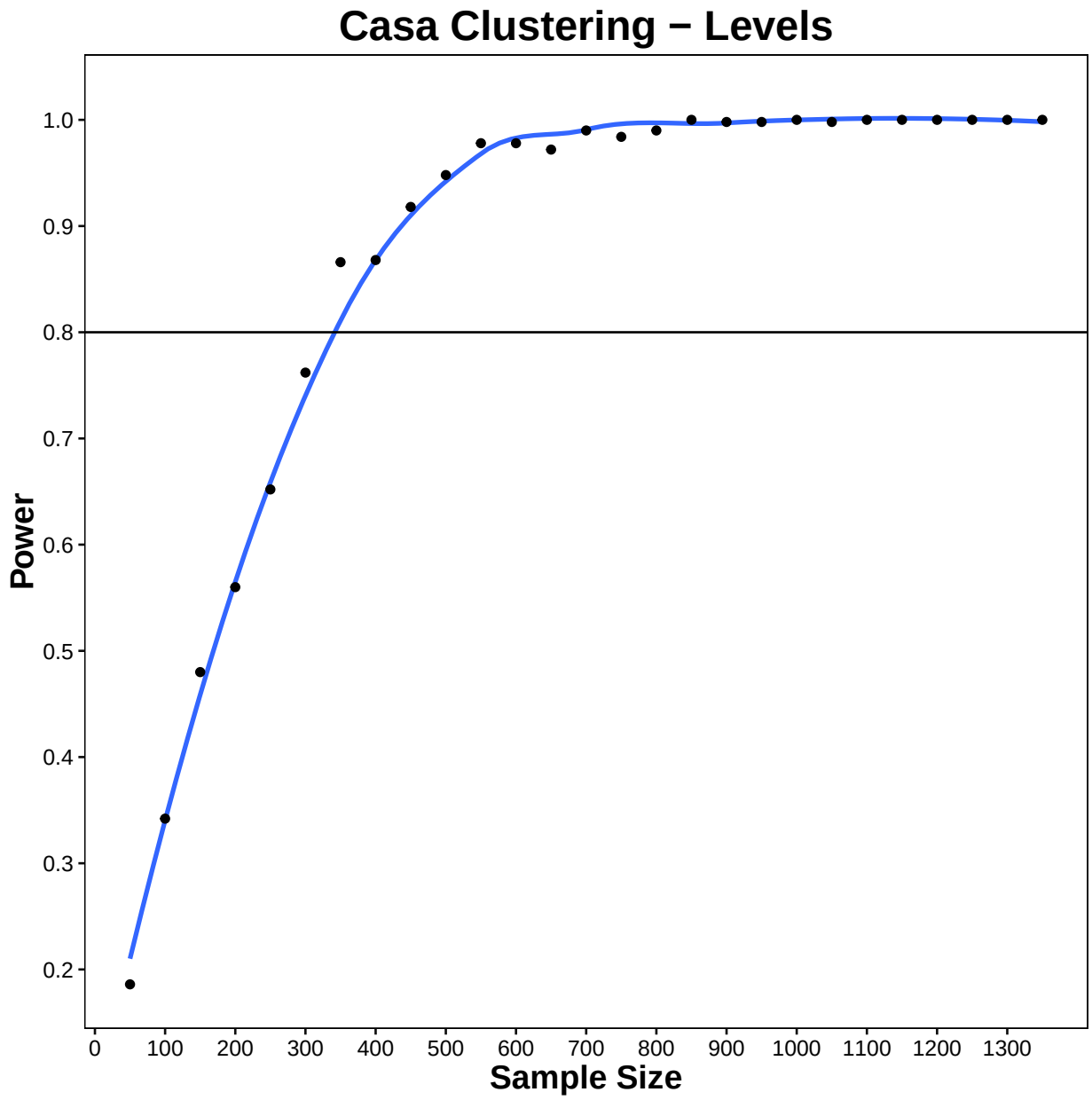
```
powerCalcSim <- function(nOrig, stepSize, simIterations,
                         clusterSwitch, levelSwitch, logSwitch){
  # create empty powerDB
  powerDB <- data.frame(sampleSize = integer(), power = double())

  # 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)
```

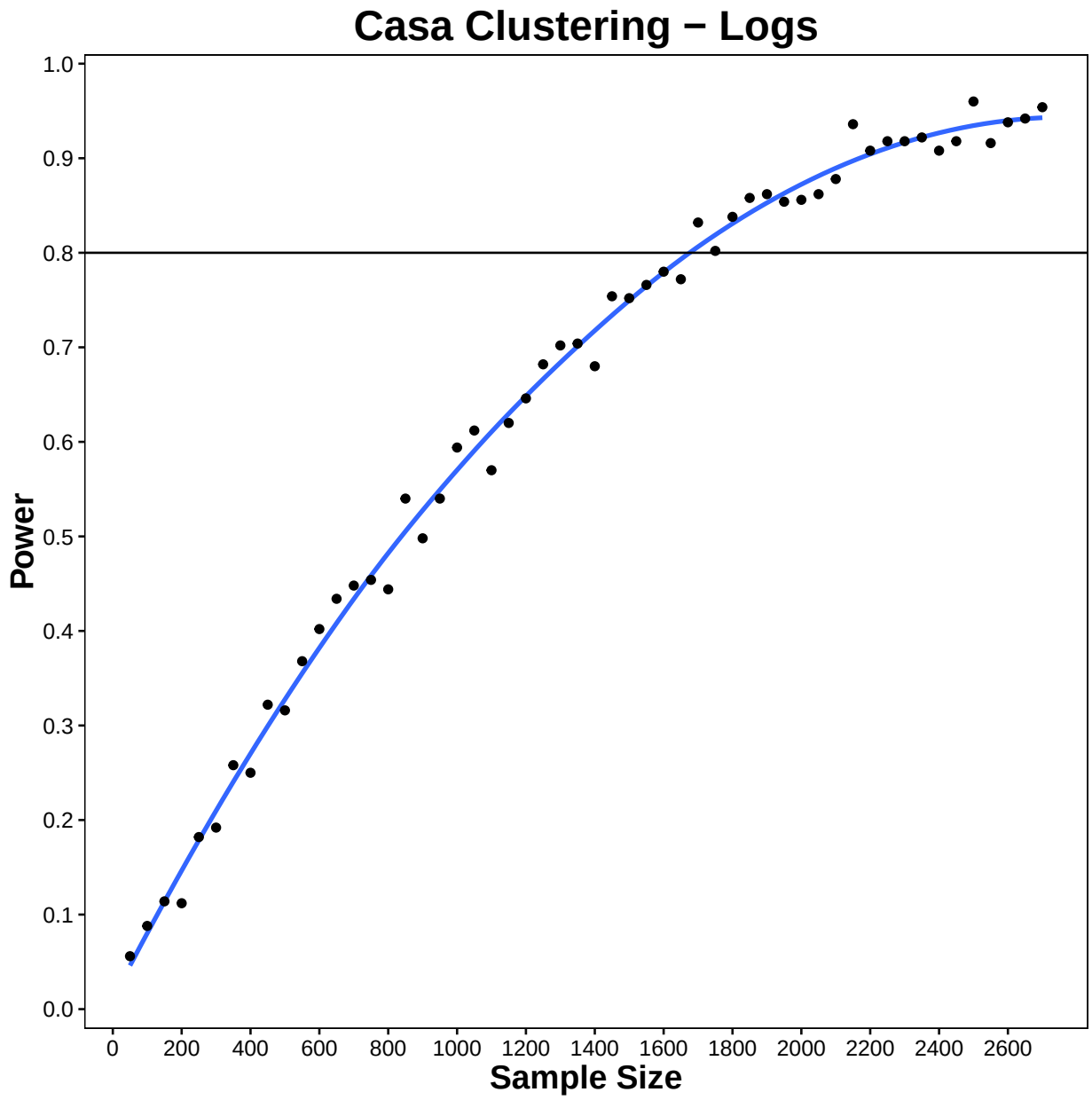
```
# export the results
write_excel_csv(pwr1, file.path(OUT,"SET1-noClusterLevels.csv"))
```

```
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(fill = NA) + 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(fill = NA) + 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(fill = NA) + 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(fill = NA) + 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
Power
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
Sample Size

```
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 perferred 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 casas
  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 <- felm(energia ~ treatedSMS | bootID + sampleMonth| 0 | 0, data = bootData)
  }
  if(cluster == FALSE & logs == TRUE){
    bootData <- bootData %>%
      mutate(logEnergia = log(energia) + MDE*treatedSMS)
   nReg <- felm(logEnergia ~ treatedSMS | bootID + sampleMonth| 0 | 0, data = bootData)
  }
  if(cluster == TRUE & levels == TRUE){
    bootData <- bootData %>%
     mutate(energia = energia + MDE*treatedSMS)
   nReg <- felm(energia ~ treatedSMS | bootID + sampleMonth| 0 | bootID, data = bootData)
  }
  if(cluster == TRUE & logs == TRUE){
    bootData <- bootData %>%
      mutate(logEnergia = log(energia) + MDE*treatedSMS)
   nReg <- felm(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()
```

```
}
```

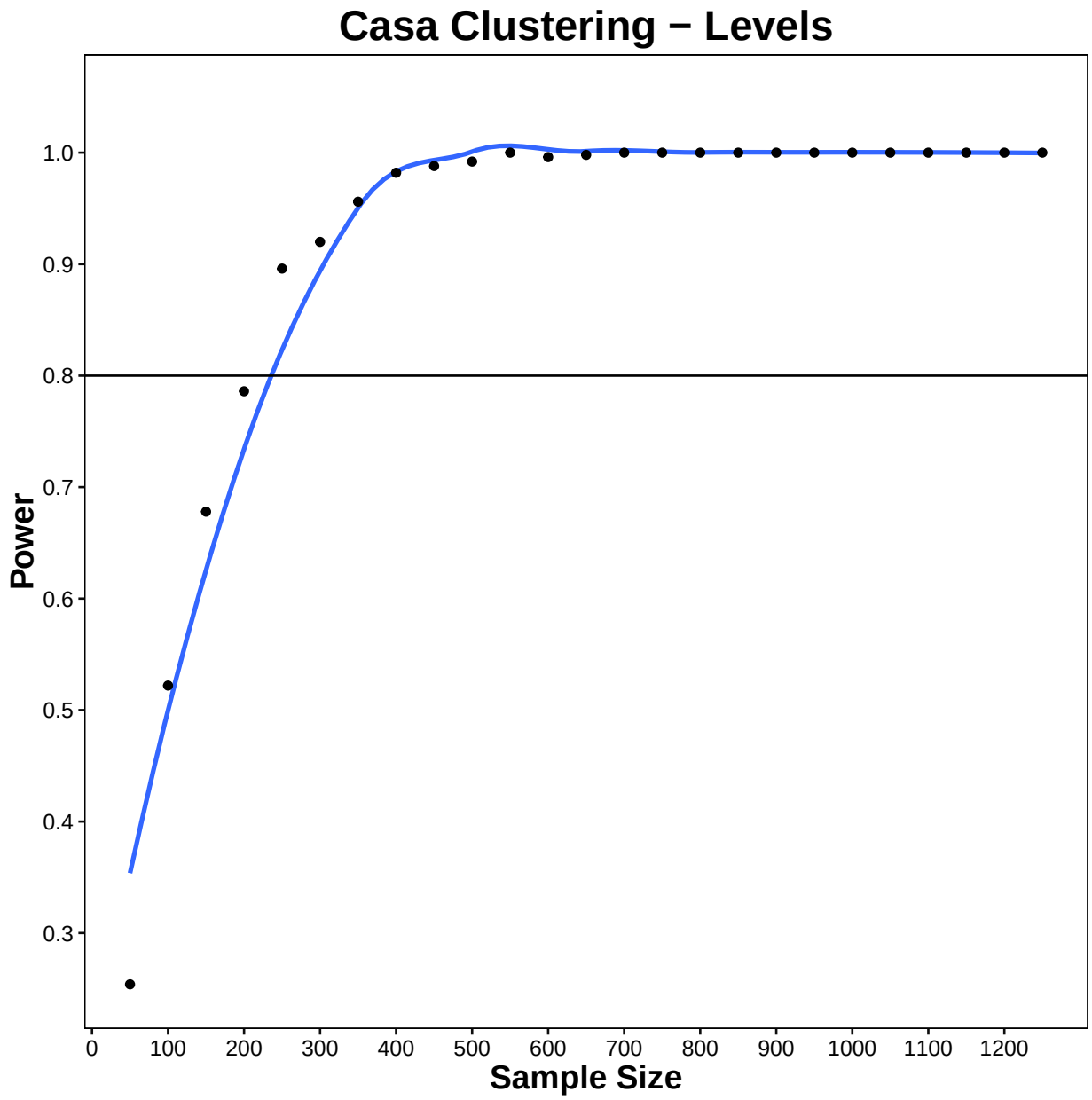
```
powerCalcSim2 <- function(nOrig, stepSize, simIterations, MDESet,
                          clusterSwitch, levelSwitch, logSwitch){
  # create empty powerDB
  powerDB <- data.frame(sampleSize = integer(), power = double())

  # 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)
```

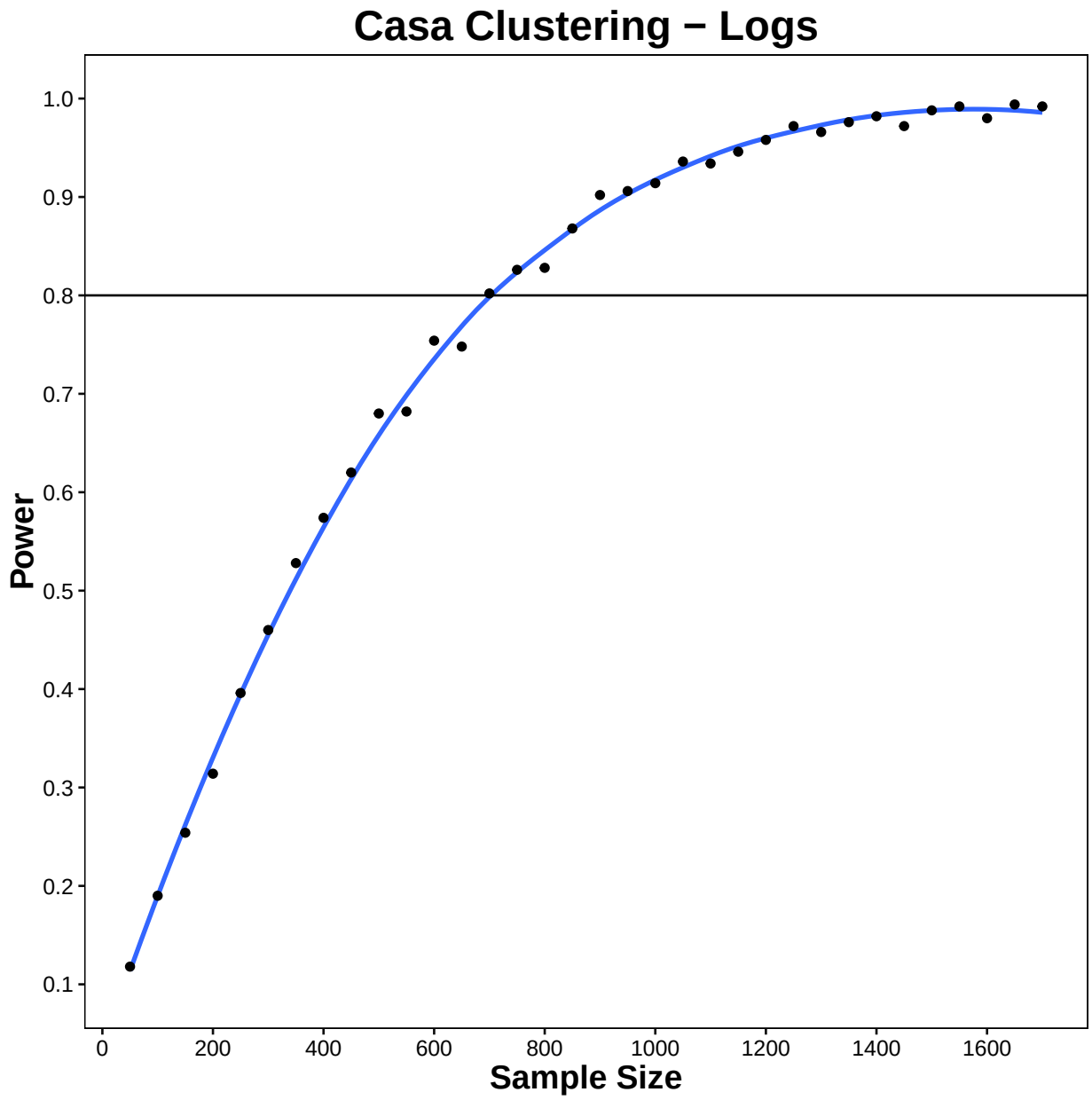
```
# export the results
write_excel_csv(pwr5, file.path(OUT,"SET2-noClusterLevels.csv"))
write_excel_csv(pwr6, file.path(OUT,"SET2-clusterLevels.csv"))
write_excel_csv(pwr7, file.path(OUT,"SET2-noClusterLogs.csv"))
write_excel_csv(pwr8, file.path(OUT,"SET2-clusterLogs.csv"))
```

```
#plot some results
plot5 <- ggplot(data = pwr5, aes(sampleSize, Power)) +
          geom_smooth(fill = NA) + 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(fill = NA) + 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(fill = NA) + 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(fill = NA) + 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
```


Casa Clustering − Levels
Power
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
Sample Size

```
plot8
```



5 Power Calculations 3

Fix Flexbox at $N = 100$ and then make control group larger

```
###################################
# SECTION 3 - POWER CALCS SIM 3
###################################

# 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 = 100, 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
  #print(tidy(nReg)$p.value < .05)
  tidy(nReg)$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 & N < 2000){
  print(N)
  power <- mean(map_lgl(rep(N,simIterations),nFinder,
                        cluster = clusterSwitch, levels = levelSwitch, logs = logSwitch))
  #print(power)
  temp <- data.frame(sampleSize = N + 100, Power = power)
  powerDB <- rbind(powerDB, temp)
  N <- N + stepSize
  }
  print("DOING EXTRA ONES NOW")
```

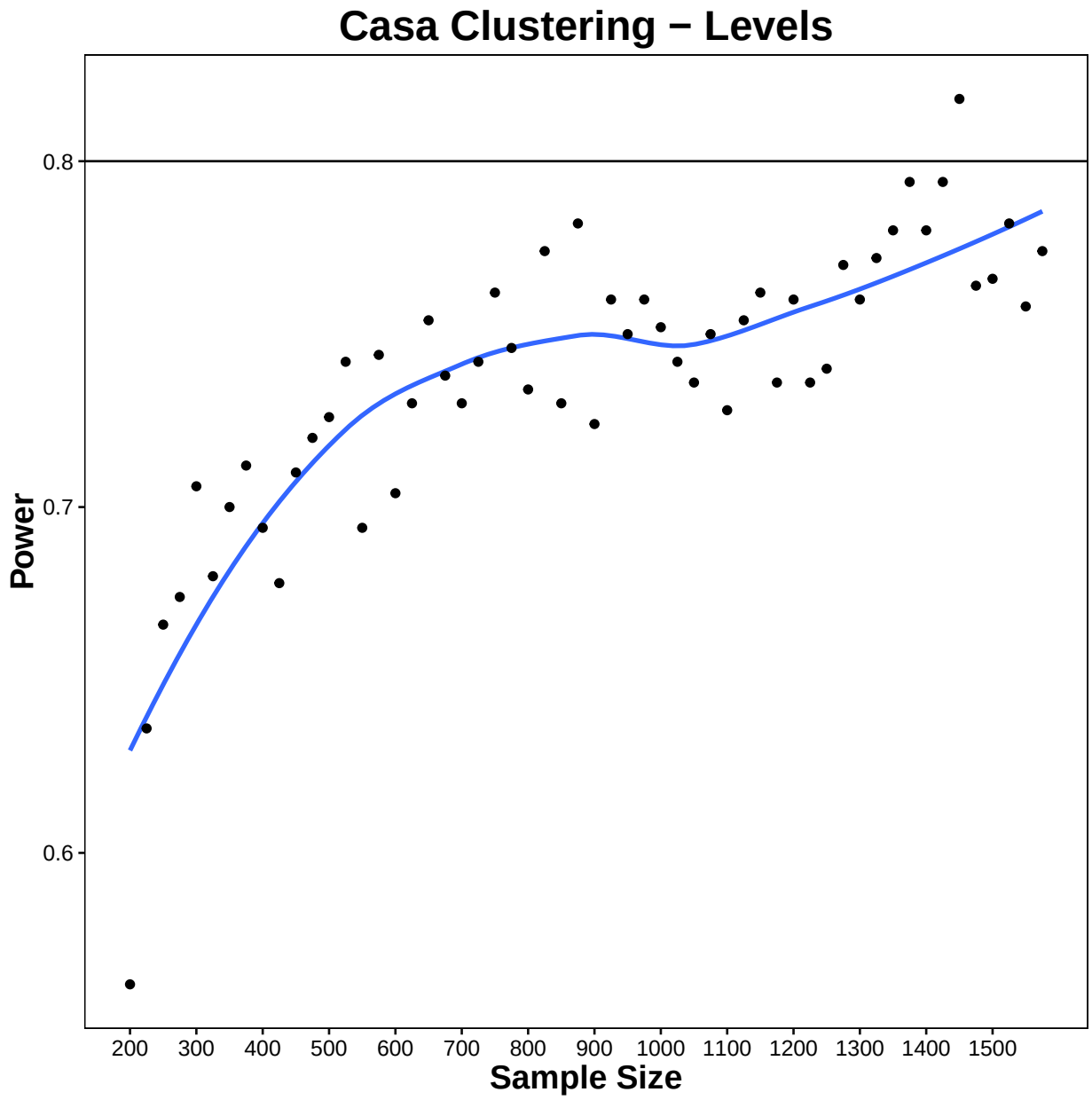
```
  for(i in 1:5){
  print(N)
  power <- mean(map_lgl(rep(N,simIterations),nFinder,
                        cluster = clusterSwitch, levels = levelSwitch, logs = logSwitch))
  #print(power)
  temp <- data.frame(sampleSize = N+100, Power = power)
  powerDB <- rbind(powerDB, temp)
  N <- N + stepSize
  }
  return(powerDB)
}
```

```
iterations <- 500
n0 <- 100
step <- 25
pwr9 <- powerCalcSim(n0,step,iterations,
                        clusterSwitch = FALSE, levelSwitch = TRUE,  logSwitch = FALSE)
pwr10 <- powerCalcSim(n0,step,iterations,
                        clusterSwitch = TRUE,  levelSwitch = TRUE,  logSwitch = FALSE)
pwr11 <- powerCalcSim(n0,step,iterations,
                        clusterSwitch = FALSE, levelSwitch = FALSE, logSwitch = TRUE)
pwr12 <- powerCalcSim(n0,step,iterations,
                        clusterSwitch = TRUE,  levelSwitch = FALSE, logSwitch = TRUE)
```

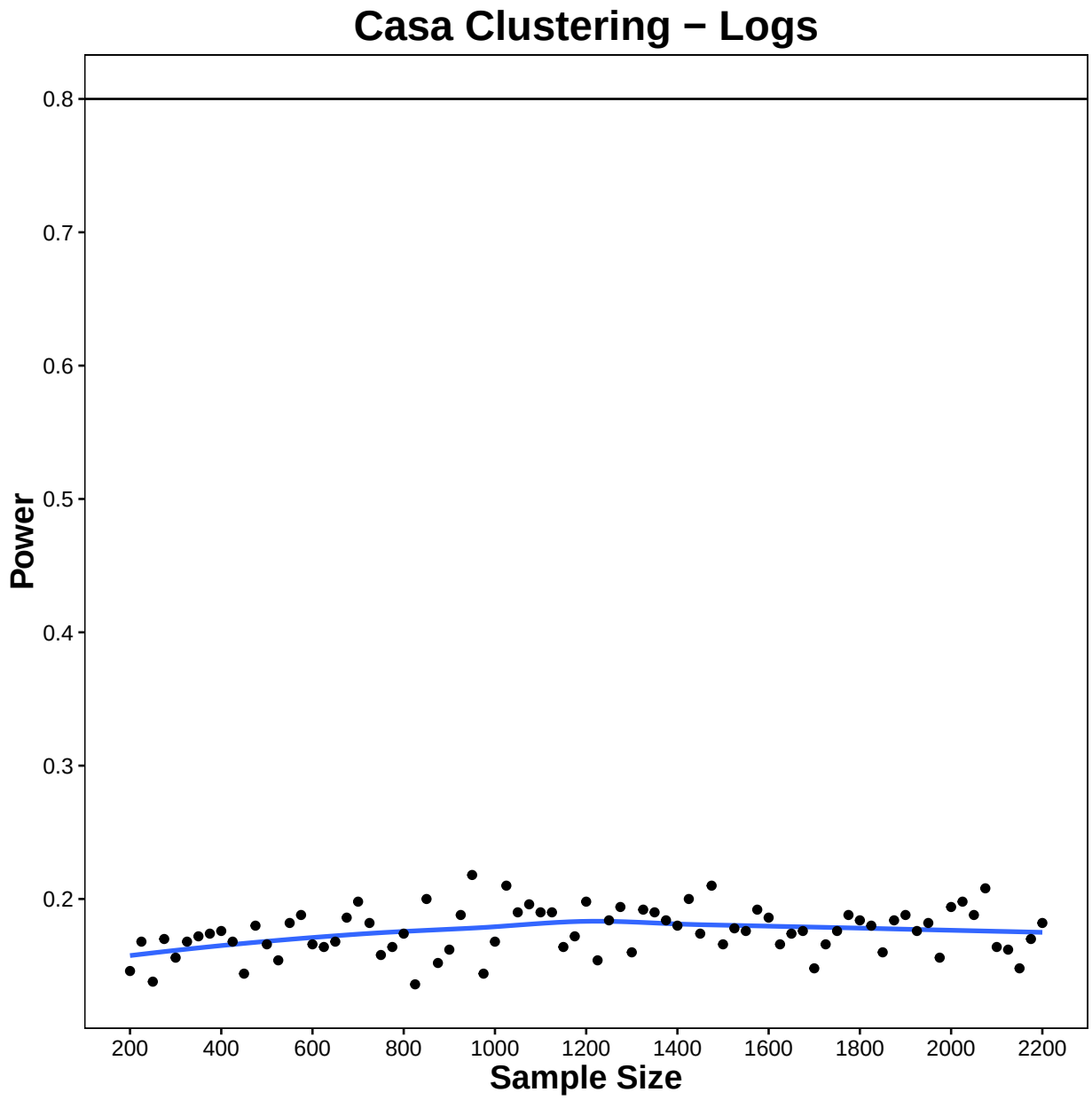
```
# export the results
write_excel_csv(pwr9, file.path(OUT,"SET3-noClusterLevels.csv"))
write_excel_csv(pwr10, file.path(OUT,"SET3-clusterLevels.csv"))
write_excel_csv(pwr11, file.path(OUT,"SET3-noClusterLogs.csv"))
write_excel_csv(pwr12, file.path(OUT,"SET3-clusterLogs.csv"))
```

```
#plot some results
plot9 <- ggplot(data = pwr9, aes(sampleSize, Power)) +
          geom_smooth(fill = NA) + geom_point() + geom_hline(yintercept=.8) +
          myGGTheme +
          ggtitle("No Clustering - Levels") +
          xlab("Sample Size") +
          scale_x_continuous(breaks = seq(0,range(pwr9$sampleSize)[2],100)) +
          ylab("Power") +
          scale_y_continuous(breaks = seq(0,1,.1))

plot10 <- ggplot(data = pwr10, aes(sampleSize, Power)) +
          geom_smooth(fill = NA) + geom_point() + geom_hline(yintercept=.8) +
          myGGTheme +
          ggtitle("Casa Clustering - Levels") +
          xlab("Sample Size") +
          scale_x_continuous(breaks = seq(0,range(pwr10$sampleSize)[2],100)) +
          ylab("Power") +
          scale_y_continuous(breaks = seq(0,1,.1))

plot11 <- ggplot(data = pwr11, aes(sampleSize, Power)) +
          geom_smooth(fill = NA) + geom_point() + geom_hline(yintercept=.8) +
          myGGTheme +
          ggtitle("No Clustering - Logs") +
          xlab("Sample Size") +
```

```
          scale_x_continuous(breaks = seq(0,range(pwr11$sampleSize)[2],200)) +
          ylab("Power") +
          scale_y_continuous(breaks = seq(0,1,.1))

plot12 <- ggplot(data = pwr12, aes(sampleSize, Power)) +
          geom_smooth(fill = NA) + geom_point() + geom_hline(yintercept=.8) +
          myGGTheme +
          ggtitle("Casa Clustering - Logs") +
          xlab("Sample Size") +
          scale_x_continuous(breaks = seq(0,range(pwr12$sampleSize)[2],200)) +
          ylab("Power") +
          scale_y_continuous(breaks = seq(0,1,.1))

plot10
```


Casa Clustering − Levels
Power
0.8
0.7
0.6
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
Sample Size

```
plot12
```



6 Simulation #4

```
# 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 = ifelse(bootID <= 100, 1,0))

  # create dataset based on sampled casas
  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 <- felm(energia ~ treatedSMS | bootID + sampleMonth| 0 | 0, data = bootData)
  }
  if(cluster == FALSE & logs == TRUE){
    bootData <- bootData %>%
      mutate(logEnergia = log(energia) + MDE*treatedSMS)
   nReg <- felm(logEnergia ~ treatedSMS | bootID + sampleMonth| 0 | 0, data = bootData)
  }
  if(cluster == TRUE & levels == TRUE){
    bootData <- bootData %>%
     mutate(energia = energia + MDE*treatedSMS)
   nReg <- felm(energia ~ treatedSMS | bootID + sampleMonth| 0 | bootID, data = bootData)
  }
  if(cluster == TRUE & logs == TRUE){
    bootData <- bootData %>%
      mutate(logEnergia = log(energia) + MDE*treatedSMS)
   nReg <- felm(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 & N < 2000){
  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
```

```
  }
  print("now the extras")
  for(i in 1:5){
  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 <- 200
step <- 25
iterations <- 200

pwr13 <- powerCalcSim2(n0,step,iterations, smsLevelEffectBalanced,
                       clusterSwitch = FALSE, levelSwitch = TRUE,  logSwitch = FALSE)
pwr14 <- powerCalcSim2(n0,step,iterations, smsLevelEffectBalanced,
                       clusterSwitch = TRUE,  levelSwitch = TRUE,  logSwitch = FALSE)
pwr15 <- powerCalcSim2(n0,step,iterations, percentEffectLogBalanced,
                       clusterSwitch = FALSE, levelSwitch = FALSE, logSwitch = TRUE)
pwr16 <- powerCalcSim2(n0,step,iterations,percentEffectLogBalanced,
                       clusterSwitch = TRUE,  levelSwitch = FALSE, logSwitch = TRUE)
```

```
# export the results
write_excel_csv(pwr13, file.path(OUT,"SET4-noClusterLevels.csv"))
write_excel_csv(pwr14, file.path(OUT,"SET4-clusterLevels.csv"))
write_excel_csv(pwr15, file.path(OUT,"SET4-noClusterLogs.csv"))
write_excel_csv(pwr16, file.path(OUT,"SET4-clusterLogs.csv"))
```

```
#plot some results
plot13 <- ggplot(data = pwr13, aes(sampleSize, Power)) +
          geom_smooth(fill = NA) + geom_point() + geom_hline(yintercept=.8) +
          myGGTheme +
          ggtitle("No Clustering - Levels") +
          xlab("Sample Size") +
          scale_x_continuous(breaks = seq(0,range(pwr13$sampleSize)[2],100)) +
          ylab("Power") +
          scale_y_continuous(breaks = seq(0,1,.1))

plot14 <- ggplot(data = pwr14, aes(sampleSize, Power)) +
          geom_smooth(fill = NA) + geom_point() + geom_hline(yintercept=.8) +
          myGGTheme +
          ggtitle("Casa Clustering - Levels") +
          xlab("Sample Size") +
          scale_x_continuous(breaks = seq(0,range(pwr14$sampleSize)[2],100)) +
          ylab("Power") +
          scale_y_continuous(breaks = seq(0,1,.1))

plot15 <- ggplot(data = pwr15, aes(sampleSize, Power)) +
          geom_smooth(fill = NA) + geom_point() + geom_hline(yintercept=.8) +
```

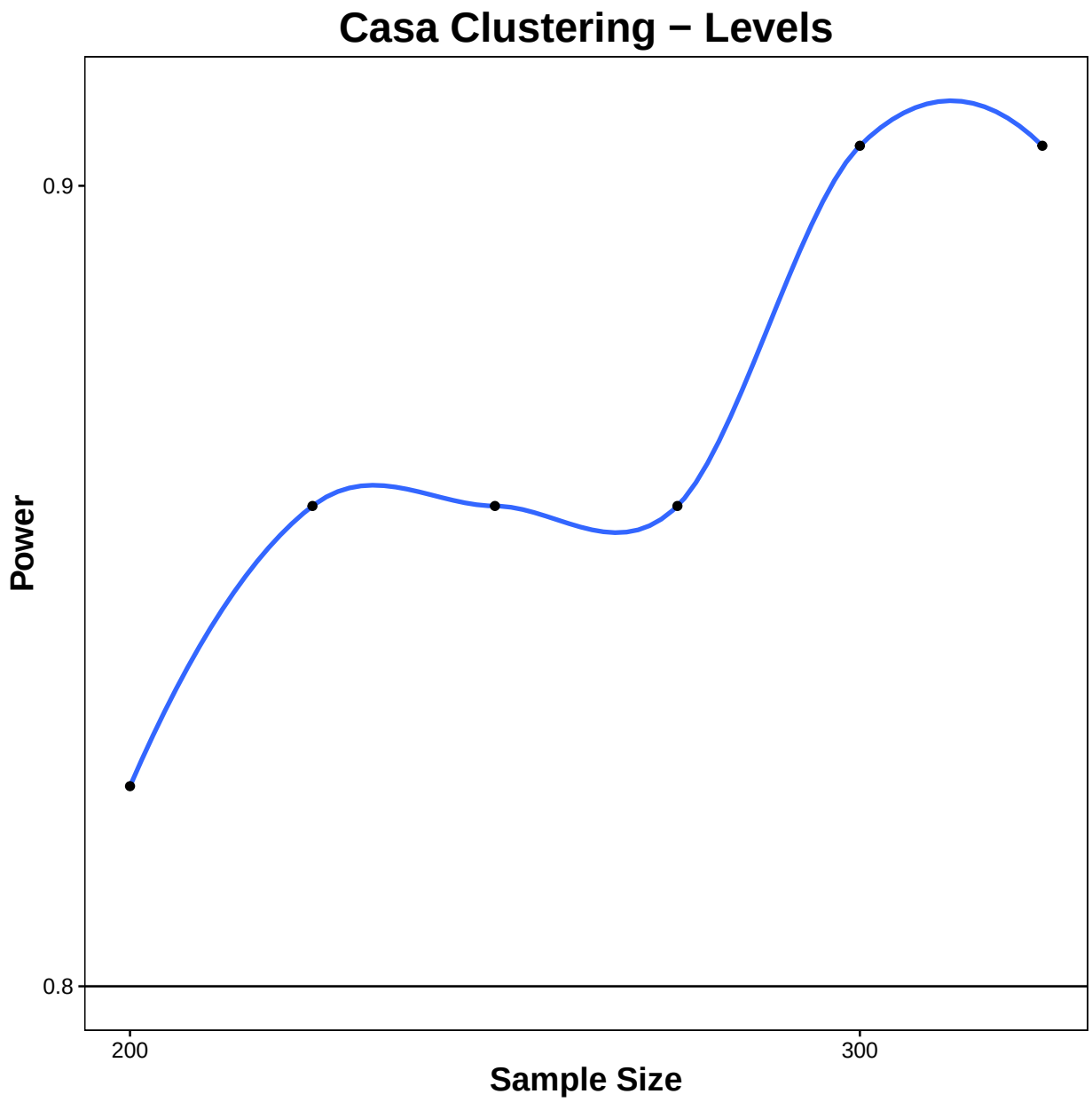
```
          myGGTheme +
          ggtitle("No Clustering - Logs") +
          xlab("Sample Size") +
          scale_x_continuous(breaks = seq(0,range(pwr15$sampleSize)[2],200)) +
          ylab("Power") +
          scale_y_continuous(breaks = seq(0,1,.1))

plot16 <- ggplot(data = pwr16, aes(sampleSize, Power)) +
          geom_smooth(fill = NA) + geom_point() + geom_hline(yintercept=.8) +
          myGGTheme +
          ggtitle("Casa Clustering - Logs") +
          xlab("Sample Size") +
          scale_x_continuous(breaks = seq(0,range(pwr16$sampleSize)[2],200)) +
          ylab("Power") +
          scale_y_continuous(breaks = seq(0,1,.1))

plot14
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, :  Chernobyl!
trL>n 6
```

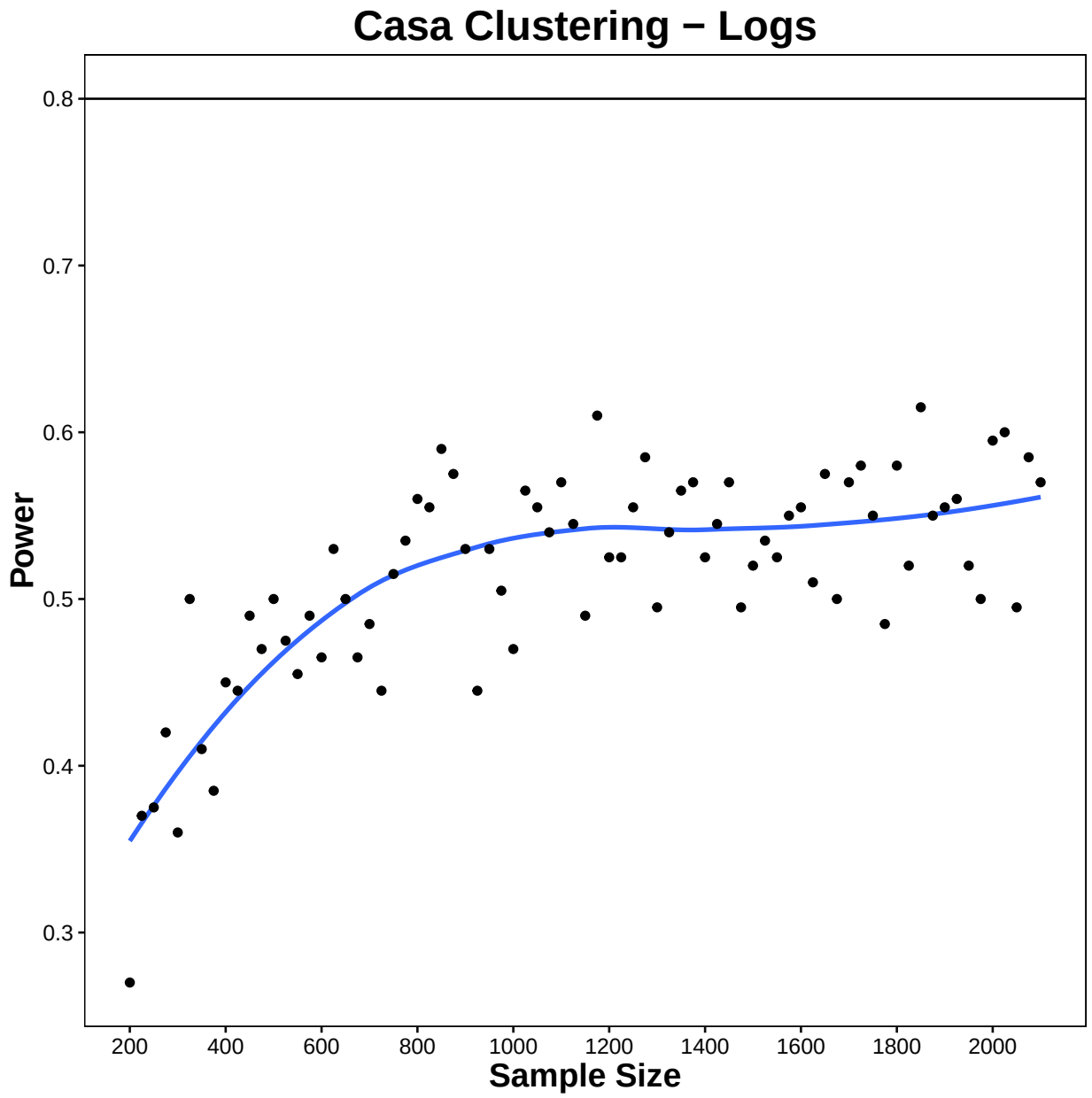
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric, :  Chernobyl!
trL>n 6
```

```
## Warning in sqrt(sum.squares/one.delta):  NaNs produced
```

```
## Warning in stats::qt(level/2 + 0.5, pred$df):  NaNs produced
```




```
plot16
```


Casa Clustering − Logs
Power
0.8
0.7
0.6
0.5
0.4
0.3
200
400
600
800
1000
1200
1400
1600
1800
2000
Sample Size