R Notebook

A bit of stats

Does eating a banana makes you happier ? Lets assume happiness is distributed following a gaussian distribution. You have a way of measuring it reliably.

- You are alone eating a banana:
  - what is the magnitude ?
  - what is the significance ?
- You are doing it with 4 friends
  - what is the magnitude ?
  - what is the significance ?
- You are doing it with 40000 people
  - Let's say you see a 3% increase in happiness
  - Would it be significant ?
- You are doing it with 40000 in china and 40000 in the US
  - Will the effect be significantly different ?
  - How will you know if that difference is really meaningful ?

```r
before_happiness = rnorm(40000,7,5)
after_happiness = before_happiness + rnorm(40000,0.5,1)
mean(after_happiness - before_happiness)
```

```
## [1] 0.4962805
```

```r
# What about
after_happiness = before_happiness + rnorm(40000,0.5,35)
```

Power Analysis

A concrete example, you want to know how many subject are necessary for your comparative experiment. You have two groups: - Group 1 is doing A then B - Group 2 is doing B then A

Research questions: - Do you need groups to test if task B is correlated to task A ? - You are wondering how many subject are necessary to show a difference in performance between the two groups on task A and B - Is this difference larger for task A or task B. - How many subject would you need if you wanted to know the exact magnitude of the effect size with precision ?

```r
# Comonly used values in clinical trial design
qnorm(0.05)
```

```
## [1] -1.644854
```

```r
pnorm(1.644854)
```

```
## [1] 0.95
```

```r
power = 0.9
false_positive_error_rate = 0.05
alpha = false_positive_error_rate

# Specific to the problem
effect_size = 0.1
```

```r
standard_deviation = 0.22
variance = standard_deviation^2

mean = 0.64
estimated_n = 364
n = estimated_n

quantile = qnorm(0.05, mean, standard_deviation)-(effect_size/(standard_deviation/sqrt(n)))
estimated_power = 1 - pnorm(quantile, mean, standard_deviation)
print(paste0("Estimated power: ", estimated_power))
```

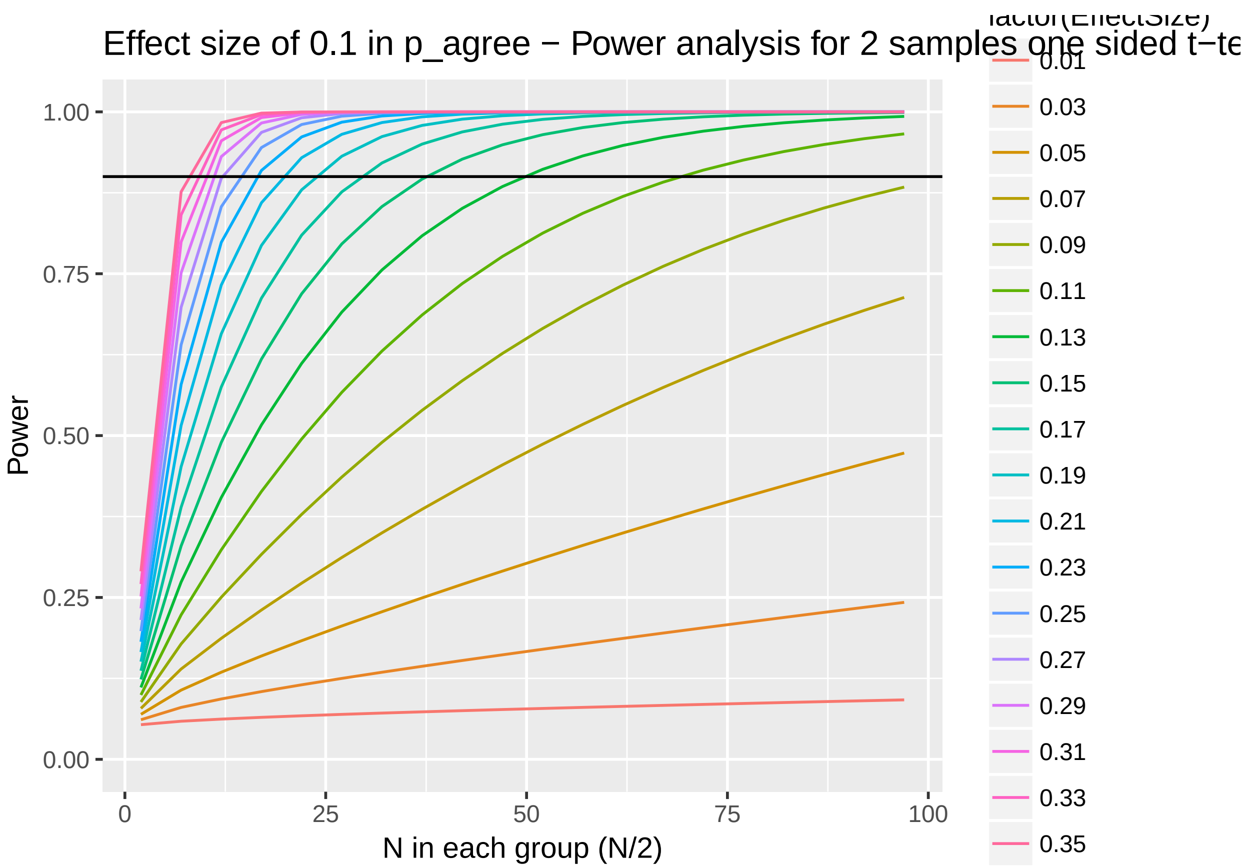
```
## [1] "Estimated power: 1"
```

```r
get_estimated_power = function (n, mean, standard_deviation, effect_size, alpha) {
  # quantile = qnorm(0.05)-(effect_size/(standard_deviation/sqrt(n)))
  # estimated_power = 1 - pnorm(quantile)
  power_test = power.t.test(n = n, delta = effect_size, sd = standard_deviation, sig.level = alpha,
              power = NULL,
              type = "two.sample",
              alternative = "one.sided")

  return(power_test$power)
}

n = seq(2,100,5)
x = y = z = NULL
d = seq(0.01,0.35,0.02)
for (delta in d) {
  x = c(x,n)
  y = c(y, rep(delta, NROW(n)))
  z = c(z, sapply(n, get_estimated_power, mean, standard_deviation, delta, alpha))
}

# sapply(n, get_estimated_power, mean, standard_deviation, effect_size, alpha)
library(scatterplot3d)
library(ggplot2)
library(data.table)

results = data.table(N=x,Power=z, EffectSize=y)
results[Power>0.9, list(Needs_Minimum_N_Subjects=min(N)), by="EffectSize"]
```

```
##     EffectSize Needs_Minimum_N_Subjects
##  1:       0.11                       72
##  2:       0.13                       52
##  3:       0.15                       42
##  4:       0.17                       32
##  5:       0.19                       27
##  6:       0.21                       22
##  7:       0.23                       17
##  8:       0.25                       17
##  9:       0.27                       17
## 10:       0.29                       12
## 11:       0.31                       12
## 12:       0.33                       12
```

```
## 13:       0.35                           12
```

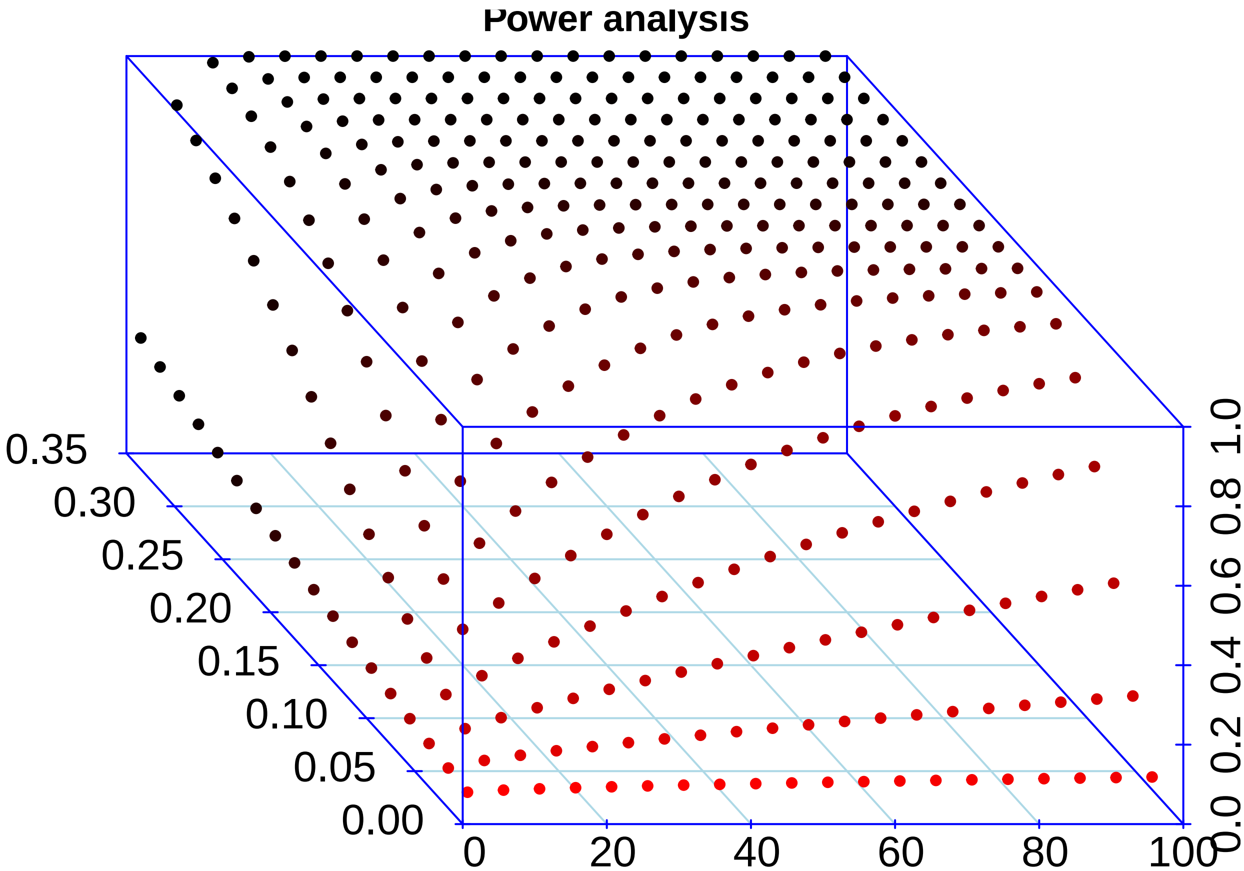
```
ggplot(data.frame(N=x,Power=z, EffectSize=y), aes(N,Power, color=factor(EffectSize), group=EffectSize))
```



Try in 3D

Is 3D a better choice to plot these curve ?

```
N = x
effect_size = y
Power = z
threshold = rep(0.9, NROW(Power))

s3d = scatterplot3d(N, effect_size, Power, highlight.3d = TRUE, angle = 120,
col.axis = "blue", col.grid = "lightblue", cex.axis = 1.3,
cex.lab = 1.1, main = "Power analysis", pch = 20, mar = c(0.5, 0.3, 0.4, 0.3))
```




```
# s3d$points3d(effect_size, N, threshold, color = "blue", type = "l", angle = 70)
```