Power Analysis

12/11/2023

```
# install.packages("pwr") need to install if you don't have it yet
library(pwr)
library(data.table)
library(sandwich)
library(lmtest)

library(ggplot2)
library(knitr)
```

1 Exploring the Influence of Personality Profiles on User Engagement in Dating Apps

1.1 Project

This study investigates the user personalities displayed on dating app profiles and their subsequent engagement levels. By comparing the effects of creating a neutral personality profile with crafting a highly humorous and intriguing personality description, we aim to examine the impact of varying personality portrayals in dating app profiles (independent of accompanying photos) on user interactions and engagement.

1.2 Samples

Our study utilized samples drawn from five distinct account profiles, each created by individual team members. Each team member generated two account profiles featuring the same photo: one with a factual description (control) and another with a humorous description (treatment). The team members, on a daily basis, swiped right on 10 candidates using the app and recorded the number of matches as our treatment effect. Consequently, our daily samples consisted of 10 profiles in the control group and 10 profiles in the treatment group, summing up to 10 * 5 profiles for each category.

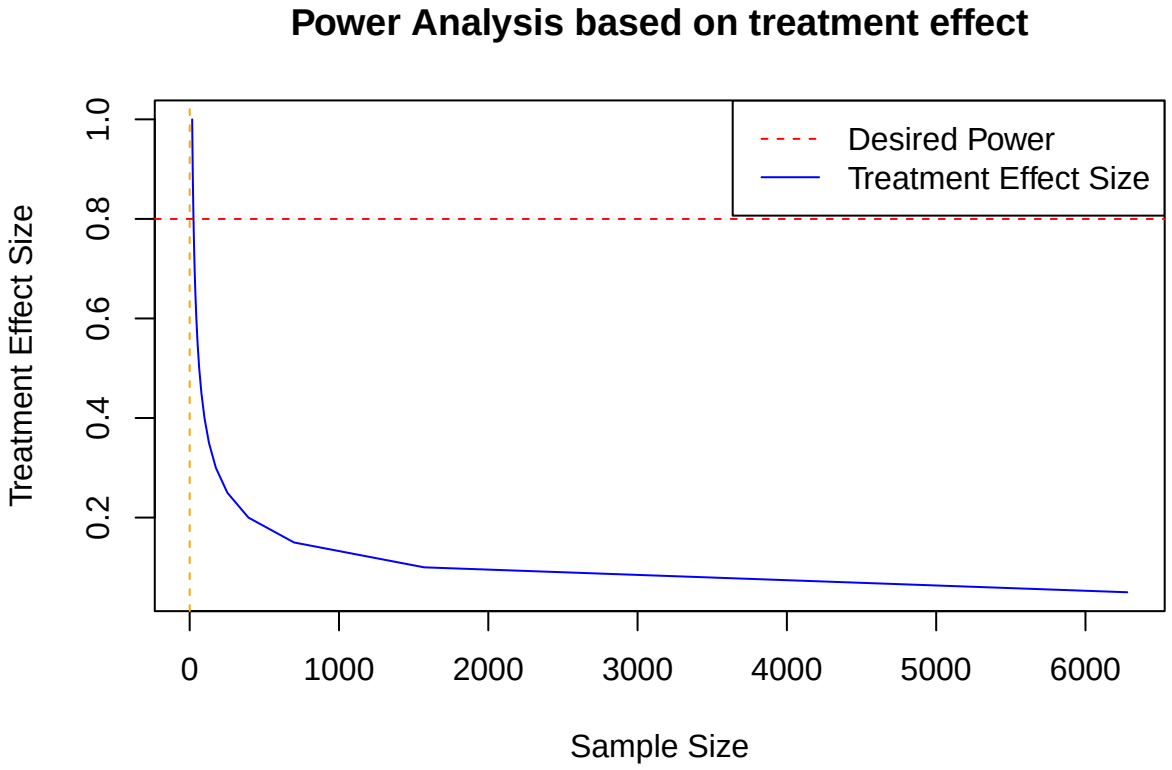
1.3 To simulate a power analysis based on treatment effect

(Need to look at the other research paper)

The provided code and plot are derived from simulating various treatment effect sizes, ranging from 5% to 100% with a 5% incremental increase, while aiming to achieve a consistent power level of 0.8. This illustrates how the necessary sample size changes with varying treatment effect sizes.

```
# Set parameters
alpha <- 0.05           # Significance level
power <- 0.8            # Desired power
treatment_effect_sizes <- seq(0.05, 1, by = 0.05)  # Range of treatment effect sizes


# Initialize variables
target_sample_size <- NA
target_effect_size <- NA
```

```r
# Function to calculate power for a given effect size and sample size
calculate_power <- function(effect_size, sample_size) {
  pwr.t.test(d = effect_size, n = sample_size, sig.level = alpha, type = "two.sample")$power
}

# Iterate through treatment effect sizes and find the combination that achieves the desired power
for (effect_size in treatment_effect_sizes) {
  # Use a range of sample sizes for exploration
  for (sample_size in seq(10, 500, by = 10)) {
    current_power <- calculate_power(effect_size, sample_size)

    if (current_power >= power) {
      target_sample_size <- sample_size
      target_effect_size <- effect_size
      break
    }
  }

  if (!is.na(target_sample_size)) {
    break
  }
}

# Function to calculate sample size for a given treatment effect size
calculate_sample_size <- function(effect_size, alpha, power) {
  pwr.t.test(d = effect_size, sig.level = alpha, power = power, type = "two.sample")$n
}

# Calculate sample sizes for each treatment effect size
sample_sizes <- sapply(treatment_effect_sizes, calculate_sample_size, alpha = alpha, power = power)

# Plot the results
plot(sample_sizes, treatment_effect_sizes, type = "l", col = "blue",
     xlab = "Sample Size", ylab = "Treatment Effect Size",
     main = "Power Analysis based on treatment effect")

# Add a horizontal line for the desired power
abline(h = power, col = "red", lty = 2)
abline(v = target_effect_size, col = "orange", lty = 2)
# Add legend
legend("topright", legend = c("Desired Power", "Treatment Effect Size"),
       col = c("red", "blue"), lty = c(2, 1))
```




```
# Print the results
cat("Desired Power:", power, "\n")
```

```
## Desired Power: 0.8
```

```
cat("Target Sample Size:", target_sample_size, "\n")
```

```
## Target Sample Size: 400
```

```
cat("Target Treatment Effect Size:", target_effect_size, "\n")
```

```
## Target Treatment Effect Size: 0.2
```

According to the analysis, in order to uphold a power level of 0.8, the treatment effect should be 20% with a sample size of 400