Power Analysis

02/16/2025

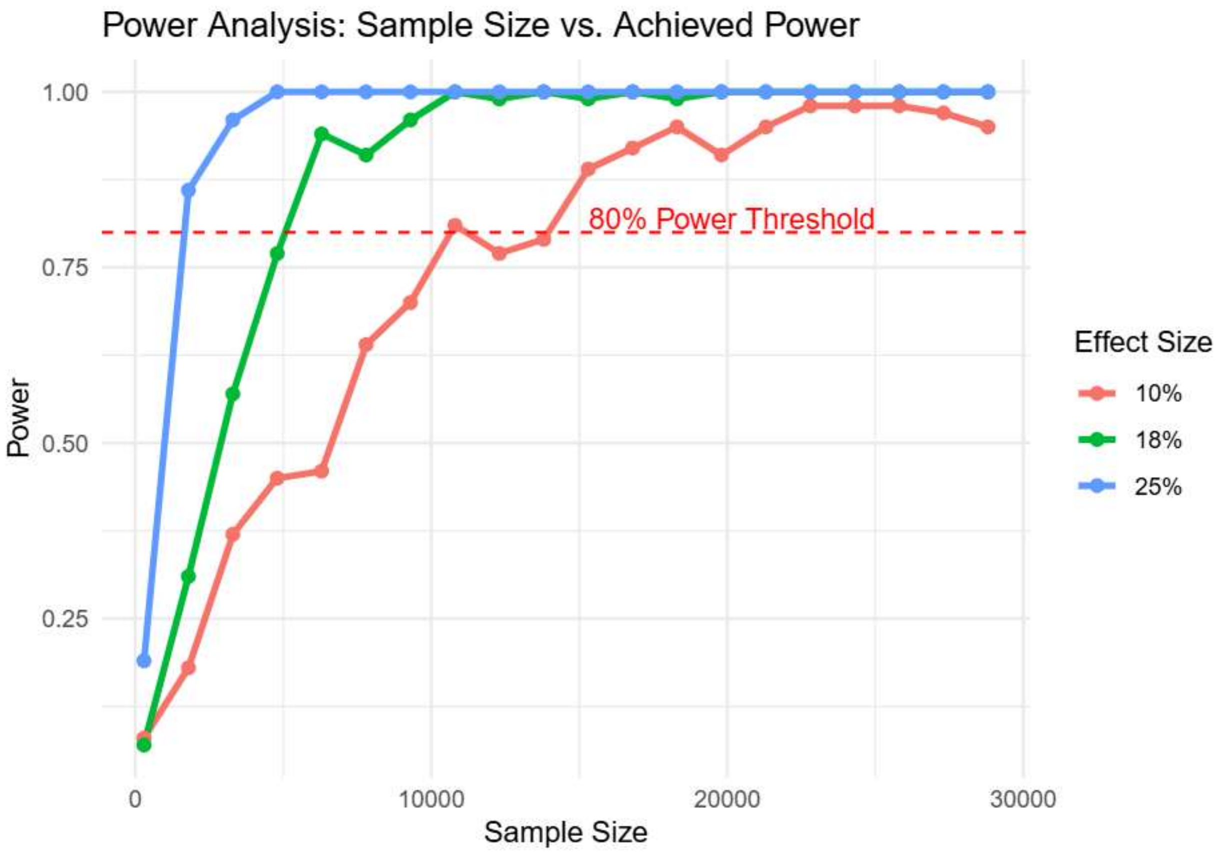
```r
library(ggplot2)

set.seed(123)

# Function to generate random user IDs
generate_user_id <- function(n) {
  replicate(n, paste0(sample(c(0:9, letters, LETTERS), 12, replace = TRUE), collapse = ""))
}

n <- 30000

# Define baseline conversion rates and lifts
baseline_rate <- 0.24  # 24% of users apply again in the same session
effect_sizes <- c(0.10, 0.18, 0.25)  # 10%, 18%, and 25% increases

# Dataframe to store power results
power_results <- data.frame()

# Nested loops for different effect sizes
for (effect_size in effect_sizes) {

  # Create the dataframe for each scenario
  df <- data.frame(
    user_id = generate_user_id(n),
    test_group = sample(c("treatment", "control"), n, replace = TRUE),
    applied = NA
  )

  # Apply baseline conversion for the control group
  df$applied[df$test_group == "control"] <- rbinom(sum(df$test_group == "control"), 1, baseline_rate)

  # Apply lift for the treatment group (baseline * (1 + effect_size))
  df$applied[df$test_group == "treatment"] <- rbinom(sum(df$test_group == "treatment"), 1, baseline_rat

  # Initialize vector to store power by sample size
  power_by_size <- numeric(length = length(seq(0.01, 1, by = 0.05)))
  percentages_to_sample <- seq(0.01, 1, by = 0.05)  # Adjust granularity

  # Loop through different sample sizes
  for (i in seq_along(percentages_to_sample)) {
    sample_size <- floor(percentages_to_sample[i] * nrow(df))  # Actual sample size

    if (sample_size == 0) next  # Skip iteration if sample size is zero

    t_test_p_values <- rep(NA, 100)  # Store p-values
```

```r
    for (j in 1:100) {
      # Ensure treatment and control sample sizes are balanced
      treatment_sample_size <- floor(sample_size / 2)
      control_sample_size <- sample_size - treatment_sample_size

      # Sample treatment and control data
      treatment_sample <- df[df$test_group == "treatment", ]
      control_sample <- df[df$test_group == "control", ]

      treatment_sample <- treatment_sample[sample(1:nrow(treatment_sample),
                                                  size = min(treatment_sample_size, nrow(treatment_samp
                                                  replace = TRUE), ]
      control_sample <- control_sample[sample(1:nrow(control_sample),
                                              size = min(control_sample_size, nrow(control_sample)),
                                              replace = TRUE), ]

      # Combine treatment and control samples
      sampled_data <- rbind(treatment_sample, control_sample)

      # Perform t-test on the sampled data
      t_test <- t.test(applied ~ test_group, data = sampled_data)
      t_test_p_values[j] <- t_test$p.value
    }

    # Calculate proportion of rejections (power) for this sample size
    t_test_rejects <- t_test_p_values < 0.05
    power_by_size[i] <- mean(t_test_rejects)
  }

  # Store power results
  power_results <- rbind(power_results, data.frame(
    Sample_Size = percentages_to_sample * n,
    Power = power_by_size,
    Effect_Size = paste0(effect_size * 100, "%")
  ))
}

# Plot power by sample size
p <- ggplot(power_results, aes(x = Sample_Size, y = Power, color = Effect_Size)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(
    title = "Power Analysis: Sample Size vs. Achieved Power",
    x = "Sample Size",
    y = "Power",
    color = "Effect Size"
  ) +
  geom_hline(yintercept = 0.8, linetype = "dashed", color = "red") +
  annotate("text", x = max(power_results$Sample_Size) * 0.7, y = 0.82,
           label = "80% Power Threshold", color = "red") +
  theme_minimal()

print(power_results)
```

```
##    Sample_Size Power Effect_Size
## 1          300  0.08         10%
## 2         1800  0.18         10%
## 3         3300  0.37         10%
## 4         4800  0.45         10%
## 5         6300  0.46         10%
## 6         7800  0.64         10%
## 7         9300  0.70         10%
## 8        10800  0.81         10%
## 9        12300  0.77         10%
## 10       13800  0.79         10%
## 11       15300  0.89         10%
## 12       16800  0.92         10%
## 13       18300  0.95         10%
## 14       19800  0.91         10%
## 15       21300  0.95         10%
## 16       22800  0.98         10%
## 17       24300  0.98         10%
## 18       25800  0.98         10%
## 19       27300  0.97         10%
## 20       28800  0.95         10%
## 21         300  0.07         18%
## 22        1800  0.31         18%
## 23        3300  0.57         18%
## 24        4800  0.77         18%
## 25        6300  0.94         18%
## 26        7800  0.91         18%
## 27        9300  0.96         18%
## 28       10800  1.00         18%
## 29       12300  0.99         18%
## 30       13800  1.00         18%
## 31       15300  0.99         18%
## 32       16800  1.00         18%
## 33       18300  0.99         18%
## 34       19800  1.00         18%
## 35       21300  1.00         18%
## 36       22800  1.00         18%
## 37       24300  1.00         18%
## 38       25800  1.00         18%
## 39       27300  1.00         18%
## 40       28800  1.00         18%
## 41         300  0.19         25%
## 42        1800  0.86         25%
## 43        3300  0.96         25%
## 44        4800  1.00         25%
## 45        6300  1.00         25%
## 46        7800  1.00         25%
## 47        9300  1.00         25%
## 48       10800  1.00         25%
## 49       12300  1.00         25%
## 50       13800  1.00         25%
## 51       15300  1.00         25%
## 52       16800  1.00         25%
## 53       18300  1.00         25%
```

```
## 54        19800  1.00          25%
## 55        21300  1.00          25%
## 56        22800  1.00          25%
## 57        24300  1.00          25%
## 58        25800  1.00          25%
## 59        27300  1.00          25%
## 60        28800  1.00          25%
```

```
print(p)
```



Key takeaways:

- If we expect a small lift (~10%) in applications, we should aim for at least 10,800 job seekers in our experiment.
- If we expect a moderate lift (~18%), a sample size of 4,800 job seekers should be sufficient.
- If we anticipate a strong effect (~25%), we can obtain reliable results with just 1,800 job seekers.