Introduction

The goal of today's design is to better understand the foundations of sequential testing and its relevance. We study this in an industrial context, where it is often referred to as A/B testing. It is often used to test software versions or advertisements by large corporations, to make their products more profitable. We will then conduct our own experiment, implementing the SPRT to analyze our own clicks and reaction times!

The Experiment

We first look at a recent case study by Netflix from 2024 that uses sequential testing for "canary testing". It selectively rolls out a new software version to try and identify bugs and whether play-delay increases in the update. Read through the blog post, trying to understand the setting and the role of sequential testing here. Also try to interpret their graphs and their analysis and other results.

Answer the following questions in full sentences.

1. Identify the treatment and control here. Why do you think these are also called A/B tests?

2. What is the reason behind choosing a sequential test over classical fixed-n tests in this particular setting? Give an example of another, non-software setting where sequential testing may be similarly appropriate.

3. Recall the statistical problems with "Peeking". Do you think the multiple testing corrections (such as Bonferroni) help correct for this? What would the Bonferroni correction be ($p = 0.05$) for peeking after each of a 100 data points?

4. Give an example of a Null and Alternate Hypothesis that may be applicable to the Netflix study. Interpret the results from the case study.

5. Recall the definitions of α and β, w.r.t Type-I and Type-II errors. Which do you think would be higher here?

Conducting Your Own Experiment

Conduct your own sequential test using reaction times! Split into pairs and collect binary outcomes of who reacts faster to the Human Benchmark reaction test. Note that you will collect these data-points sequentially and continue the analysis in real time.

Keep the following questions in mind, and write answers explaining your choice in each.

6. Choose the null and alternate hypothesis (p_0 and p_1) that you plan to investigate and explain why. How does this choice affect the duration of the experiment?

(Hint: What is the interpretation of $p = 0.5$ here?)

7. Choose α and β, i.e. your thresholds. Understand the reasons for both higher and lower values. Are you going to keep them equal?

8. Is this experiment random at all? If yes, what is random?

9. Do you think the runs of the experiment truly are independent? Do you think the distribution of each person's reaction time changes with longer durations? Would this affect your choice of the earlier factors?

10. Real data is often messy. How would you deal with somebody clicking too early in the reaction test? Do you discard the data or code it in a different way?

11. Compare group sequential tests and the SPRT in this setting: which do you think would practically be easier? What group size would you choose for a group sequential test here and why?

Data Analysis

We will now analyze our own collected data. You will be collecting binary data with each iteration of the experiment, and you can simply append the outcome (1 or 0) to the `my_data` vector below. Also remember to load libraries like `tidyverse` or `ggplot2` if you plan to use them! You may also store your data in a `.csv` file, and append that and reload it if you prefer.

```
# load your collected data into R

my_data <- c()
```

The SPRT

12. We shall first implement the Sequential Probability Ratio Test (SPRT) for the collected response time data in real time. Identify the assumptions, define the parameters and add your data as you collect them until the SPRT terminates!

Q: What assumptions are required by the SPRT test in this binary setting?

A:

First,

```
# define the parameters

#p0 <-      # acceptable defect rate under H0
#p1 <-      # unacceptable defect rate under H1
#alpha <-
#beta  <-
```

Calculate the log-likelihood ratio for this Bernoulli case.

```
# Calculate the formula for the log-likelihood ratio

log_lr_function <-
```

```
# Append your data (Cell 1) in real time and calculate the cumulative log-LR

log_lr <- cumsum(log_lr_function(my_data))
```

```
# SPRT thresholds
A <- (1 - beta) / alpha
B <- beta / (1 - alpha)
logA <- log(A)
logB <- log(B)
```

Check and visualize the results of your SPRT with each iteration of data below!

```
# Check results

current_log_lr <- tail(log_lr, n = 1)

# Check the SPRT stopping conditions
```

```
if (current_log_lr >= logA) {
  print("Decision Reached: Crosses Upper Boundary (logA).")
  print("Conclusion: Reject the Null Hypothesis (H0).")
} else if (current_log_lr <= logB) {
  print("Decision Reached: Crosses Lower Boundary (logB).")
  print("Conclusion: Fail to reject the Null Hypothesis (H0).")
} else {
  print("No Decision Yet: The statistic is still between the boundaries.")
  print("Conclusion: Continue collecting data!")
}
```

```
# Plot the SPRT

# Load required library
library(ggplot2)

# Create the dataframe
df_sprt <- data.frame(
  n = 1:length(log_lr),
  log_lr = log_lr
)

# Generate the plot
ggplot(df_sprt, aes(x = n, y = log_lr)) +
  geom_line(color = "black", linewidth = 0.8) +
  geom_point(color = "black", size = 1.5) +
  geom_hline(yintercept = logA, color = "black", linetype = "dashed", linewidth = 0.8) +
  geom_hline(yintercept = logB, color = "black", linetype = "dashed", linewidth = 0.8) +
  labs(
    title = "One SPRT path",
    x = "Number tested",
    y = "Log likelihood ratio"
  ) +
  theme_minimal()
```

Group Sequential Tests

We shall now implement a grouped SPRT test with the group sizes that you decided on earlier. Use the same data that you had collected but instead batch it in groups.

13. Do you think any of the assumptions of the SPRT changes in this grouped setting?

Define the group size below and get started with implementing the Grouped SPRT!

```
# Define group size and import groups one at a time

group_size =

# Calculate how many full groups we can form from our collected data
num_groups <- floor()   ## Complete the floor function!

# Extract the indices corresponding to the end of each group
group_indices <- seq(from = group_size, to = num_groups * group_size, by = group_size)
```

The log-likelihood ratio for a single observation still stays the same, but we just add `group_size` many terms to the cumulative sum at a time together and visualize it!

```
# Calculating the grouped log_lr for each group with fixed size.
# Note, ith element of grouped_log_lr represents

grouped_log_lr <- log_lr[group_indices]   ## Why does this work?
```

We now generate the same plot for the Grouped SPRT case. Note that in such a grouped setting, there are more efficient ways of selecting the thresholds `logA` and `logB`, which can be done via the `gsDesign` R package. However, for the purposes of this lab, we shall stick to the earlier standard thresholds calculated.

```
# Plot the SPRT

# Create the dataframe. Complete the steps!

df_grouped_sprt <- data.frame(
  #n = 1:length(),
  #
)

# Generate the plot. Complete the aes!

ggplot(df_grouped_sprt, aes(x = n, y = )) +
  geom_line(color = "#B22222", linewidth = 0.8) +
  geom_point(color = "#B22222", size = 1.5) +
  geom_hline(yintercept = logA, color = "black", linetype = "dashed", linewidth = 0.8) +
  geom_hline(yintercept = logB, color = "black", linetype = "dashed", linewidth = 0.8) +
   labs(
     title = "Grouped SPRT path",
     x = "Number tested",
```

```
    y = "Grouped Log likelihood ratio"
  ) +
  theme_minimal()
```

14. Compare both the SPRT and the Grouped SPRT plots above. Which stopped faster? Which seemed to converge more convincingly?