

Worksheet 04

Jan Hohenheim

2024-04-08

14/15

Congratulations!

Total (1-4) = 47/50 = 94% > 75%

Recommendation:
Take the written exam

```
library(tidyverse)
library(rjags)
library(coda)
library(bayesmeta)
library(pCalibrate)
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
## "ndiMatrix" of class "replValueSp"; definition not updated
```

```
library(ggthemes)
```

```
## Warning: package 'ggthemes' was built under R version 4.3.3
```

```
library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.3.3
```

```
library(glue)
```

```
theme_set(theme_solarized_2())
```

Exercise 1

4/5

a)

```
set.seed(42)
```

```
n_secukinumab <- 20
```

```
n_placebo <- 5
```

```
p_secukinumab <- 0.6
```

```
p_placebo <- 0.25
```

```
n_sim <- 1e6
```

```
poc <- function(n_secukinumab, n_placebo, p_secukinumab, p_placebo, n_sim){
  x_secukinumab <- rbinom(n_sim, n_secukinumab, p_secukinumab)
  x_placebo <- rbinom(n_sim, n_placebo, p_placebo)
```

```
  p_hat_secukinumab <- x_secukinumab / n_secukinumab
```

```
  p_hat_placebo <- x_placebo / n_placebo
```

```
  mean(p_hat_secukinumab > p_hat_placebo)
```

```
}
```

```
result.poc <- poc(n_secukinumab, n_placebo, p_secukinumab, p_placebo, n_sim)
```

It is a classical approximation. Please use the approach to compute the MCse you developed in the 03worksheet.

```
result.poc_se <- sqrt(result.poc * (1 - result.poc) / n_sim)
"POC: {result.poc} (MCSE: {result.poc_se})" |> glue()
```

```
## POC: 0.91773 (MCSE: 0.000274775630469662)
```

The POC is 91.8% with an MCSE of 0.0275%. This is above the 90% threshold with a pretty good precision.

b)

```
n_max_placebo <- 10
n_sim <- 1e6
randomization_ratio <- 4/1

simulations <-
  tibble(
    n_placebo = seq_len(n_max_placebo),
    n_secukinumab = seq_len(n_max_placebo) * randomization_ratio
  ) |>
  mutate(n_total = n_secukinumab + n_placebo)

"Simulated pairs following 4:1 randomization ratio" |> message()
```

```
## Simulated pairs following 4:1 randomization ratio
```

```
simulations
```

```
## # A tibble: 10 x 3
##   n_placebo n_secukinumab n_total
##   <int>      <dbl>      <dbl>
## 1         1         4         5
## 2         2         8        10
## 3         3        12        15
## 4         4        16        20
## 5         5        20        25
## 6         6        24        30
## 7         7        28        35
## 8         8        32        40
## 9         9        36        45
## 10        10        40        50
```

```
simulations <- simulations |>
  mutate(
    poc = map2_dbl(
      n_secukinumab,
      n_placebo,
      \(n_s, n_p) poc(n_s, n_p, p_secukinumab, p_placebo, n_sim))
  )
```

```
simulations |>
  filter(poc > 0.9) |>
  mutate(
    n_total = n_secukinumab + n_placebo,
    se = sqrt(poc * (1 - poc) / n_sim)
  ) |>
  arrange(n_total)
```

5/5

```
## # A tibble: 6 x 5
##   n_placebo n_secukinumab n_total   poc     se
##   <int>      <dbl>    <dbl> <dbl>  <dbl>
## 1         5         20      25 0.918 0.000274
## 2         6         24      30 0.938 0.000241
## 3         7         28      35 0.953 0.000211
## 4         8         32      40 0.964 0.000186
## 5         9         36      45 0.972 0.000164
## 6        10         40      50 0.979 0.000143
```

As we can see, the smallest number of patients needed with a 4:1 randomization ratio is indeed 25 patients in total, with 20 patients on secukinumab and 5 patients on placebo.

c)

5/5

High Level In broad terms, the approach is to simulate the trial data for different patient numbers and calculate the POC for each simulation. The POC is a percentage of how often the treatment group fares better than the placebo group in a given simulation. The higher the POC, the better. Our cutoff here is 90%, meaning that we want to see the treatment group outperform the placebo group in at least 90% of the simulations.

To find the lowest number of patients needed, we simulate different patient numbers and calculate the POC for each combination. We then see which patient numbers are the lowest that yield a POC > 90%.

We don't simulate every combination of numbers imaginable, but only those with a 4:1 randomization ratio. Baeten et al. (2013) give the following reasoning for this choice:

A 4:1 randomization ratio was chosen to reduce the number of placebo- treated patients while maintaining a double-blinded study design and to allow a limited study size in the absence of knowledge of the risk/benefit of secukinumab in ankylosing spondylitis (AS).
- Baeten et al., Supplementary appendix, 2013

This means that we will look at the POC for 1 placebo : 4 secukinumab patients, 2:8, 3:12, etc.

In-Depth For a more in-depth explanation, we can look at the code used.

First, let's go over how the POC is calculated:

```
poc <- function(n_secukinumab, n_placebo, p_secukinumab, p_placebo, n_sim){
  x_secukinumab <- rbinom(n_sim, n_secukinumab, p_secukinumab)
  x_placebo <- rbinom(n_sim, n_placebo, p_placebo)

  p_hat_secukinumab <- x_secukinumab / n_secukinumab
  p_hat_placebo <- x_placebo / n_placebo

  mean(p_hat_secukinumab > p_hat_placebo)
}
```

The algorithm can be put into three steps: 1. Simulate trial data for the secukinumab and placebo groups 1'000'000 times. 2. Calculate the response rates for both groups. 3. Calculate how often on average the response rate in the secukinumab group is higher than in the placebo group.

Now we prepare the simulated patient numbers:

```
n_max_placebo <- 10
n_sim <- 1e6
randomization_ratio <- 4/1

simulations <-
```

```
tibble(
  n_placebo = seq_len(n_max_placebo),
  n_secukinumab = seq_len(n_max_placebo) * randomization_ratio
) |>
mutate(n_total = n_secukinumab + n_placebo)
```

This creates a table as follows:

n_placebo	n_secukinumab	n_total
1	4	5
2	8	10
3	12	15

etc.

The `n_secukinumab` is a multiple of `n_placebo`, following the 4:1 randomization ratio.

This means we will simulate all 4:1 patient ratios up to 40:10 (maximum chosen arbitrarily).

For the actual simulation, we have the following code:

```
simulations <- simulations |>
mutate(
  poc = map2_dbl(
    n_secukinumab,
    n_placebo,
    \(n_secukinumab, n_placebo) poc(n_secukinumab, n_placebo, p_secukinumab, p_placebo, n_sim)))
```

For each of our simulated patient numbers, we will calculate the POC as described earlier based on 1'000'000 trials per simulation. This means we will be able to see the POC that we can expect for each patient number combination that follows a 4:1 randomization ratio.

Finally, we can filter for the patient numbers that fulfill the condition $\text{POC} > 90\%$ and sort them by the total number of patients:

```
simulations |>
filter(poc > 0.9) |>
mutate(
  n_total = n_secukinumab + n_placebo,
  se = sqrt(poc * (1 - poc) / n_sim)
) |>
arrange(n_total)
```

Per the output, we see that the smallest number of patients needed with a 4:1 randomization ratio is 20:5 if we want to achieve a $\text{POC} > 90\%$. Increasing the number of participants would further improve the POC.