Code

AMMS Practical: Introduction to Statistical Power

Bob Verity

August 04, 2022

Dependencies for Practical

Please copy and paste the below code chunk in it's entirety to your console to download R package libraries needed for this practical. If you are having trouble installing any of the R packages, please ask an instructor for a pre-loaded flash drive.

Hide

```
if (!("tidyverse" %in% installed.packages())) {
  install.packages("tidyverse")
}
```

Now load all of those libraries into this session using the code chunk below. Please copy and paste it in its entirety.

Hide

```
library(tidyverse)
```

Finally, source the additional functions that are needed for this practical by copy-pasting this function:

Hide

```
source("source_functions/power1_utils.R")
```

Intro to statistical testing and power analysis

Very often in molecular surveillance we are interested in answering simple and well-defined questions, such as:

- Has the prevalence of a drug resistance mutation increased in my population in the last 5 years?
- Has the use of bednets had an impact on parasite genotic diversity?
- Is the incidence of false-negative RDT results higher in one population than another?

We could simply measure these quantities and report our results, which is sometimes called **descriptive statistics**, but often we want more than this. We want to be able to "prove" that an effect is real. This means accounting for the role of chance in our results through **statistical testing**, sometimes called **null hypothesis testing**. If our observed results are very unlikely to happen by chance alone then it gives us more confidence that an effect is real. This does not quite "prove" that the effect is real, as we could always have been very lucky or unlucky, but it does at least control how often we come to the wrong conclusion.

Once we have a firm handle on statistical testing, the next crucial idea to grasp is **statistical power**. We can think of **power analysis** as a sort of statistical testing that we do before seeing any data. Instead, we make assumptions about the strength of the effect in the real world (for example, the difference in drug resistance prevalence) and our sample size, and we calculate *exactly* how likely we are to detect this real effect with our chosen significance test. If power is low then we are likely to miss interesting results even if they are there, because our statistical test cannot rule out the possibility that our results are down to pure chance. It is therefore critically important that we carry out power analysis before undertaking any serious trial or survey to ensure that we have a decent chance of success.

Overview of Data

For this practical we will work entirely with made-up, or simulated, datasets. These will allow us to get to grips with the basic concepts, which we can then apply to real-world datasets at a later stage.

Practical Goals

By the end of this practical, you should be able to:

- Construct a 95% confidence interval
- Carry out a t-test to compare two means, and a z-test to compare two proportions
- Calculate statistical power of several tests
- Interpret power curves and calculate optimal sample sizes
- Adjust sample sizes for expected dropout

- Comment on issues of under-powered and over-powered studies
- Design a simple study, taking account of statistical and logistical considerations

Comparing COI between two populations

Statistical testing

You have been asked to analyse some data on complexity of infection (COI) in two populations, one of which experienced a rapid scale-up of bednets and the other acting as a control. COI tends to be higher in populations with high transmission intensity, so the hypothesis here is that bednets will have caused a drop in COI on average compared to the control population.

Let's load the COI data and have a quick look at the distribution:

Hide

```
# load COI data
load("data/COI_control.RData")
load("data/COI_nets.RData")

COI_control
COI_nets
```

```
##  [1] 1 2 2 3 1 3 4 2 2 1
##  [1] 1 1 2 1 2 1 2 4 1 2
```

We can see that we have only 10 samples from each population, and it's difficult to tell from looking whether COI is greater in one population or the other.

Q1. What is the mean COI in each population? What is the variance in each population? Is the mean higher or lower in the population with bednets?

Click For Answer

A1. Means and variances shown below. The mean is slightly higher in the control group.

Hide

```
# get mean and variance of control population
mean(COI_control)
var(COI_control)
```

```
## [1] 2.1
## [1] 0.9888889
```

Hide

```
# get mean and variance of bednets population
mean(COI_nets)
var(COI_nets)
```

```
## [1] 1.7
## [1] 0.9
```

The mean COI in the control sample is 2.1. This is our best *estimate* of the mean COI in the control *population*, but we would not be at all surprised if the population value differed from 2.1 slightly. For example, it may be that the mean COI in the population is 2.0, and we just happened to sample individuals with slightly higher COIs by chance.

We can represent our uncertainty in the mean estimate through the *standard error*. The formula for the standard error is:

$$SE = \sqrt{\frac{s^2}{n}}$$

where s^2 is the sample variance and n is the sample size.

Q2. What is the standard error of the mean for the control population? What is the standard error of the mean for the bednets population?

Click For Answer

A2. We can calculate standard errors using the following code:

Hide

```
# get standard error of mean COI in the control population
SE_control <- sqrt( var(COI_control) / 10 )
SE_control

# get standard error of mean COI in the bednets population
SE_nets <- sqrt( var(COI_nets) / 10 )
SE_nets
```

```
## [1] 0.314466
## [1] 0.3
```

We can use the standard error to calculate a *95% confidence interval*. The interpretation of a confidence interval under a frequentist definition can be confusing. It states that if we were to draw samples from the same population many times (or repeat this "experiment" many times), we would expect that 95% of our calculated confidence intervals would contain the population mean. A more straightforward interpretation is that we are *95% confident that our interval contains the population mean*. Confidence intervals (CIs) are a useful way of visualising uncertainty in an estimate. The formula for a (normal) 95% CI is:

$$\bar{x} \pm 1.96 \times SE$$

where $\bar{x}$ is the sample mean.

Q3. What is the 95% CI for the control population? What is the 95% CI for the bednets population?

Click For Answer

A3. We can calculate CIs using the following code:

Hide

```
# calculate 95% CI for control population
mean(COI_control) + c(-1.96, 1.96)*SE_control

# calculate 95% CI for bednets population
mean(COI_nets) + c(-1.96, 1.96)*SE_nets
```

```
## [1] 1.483647 2.716353
## [1] 1.112 2.288
```

Q4. Do the CIs for the two population overlap? What does this tell you about how confident we are in any differences between the means?

Click For Answer

A4. Yes, they overlap. This tells us that we are not very confident about a difference in the means, because the true mean in the bednets population might be the same or even higher than the control population.

We will compare the two means using the two-sample Student's t-test. We have the same number of samples in both groups, which makes life a bit easier, and we will also assume that the variances are the same between groups.

The formula for the test statistic is as follows:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\hat{s}_1^2 + \hat{s}_2^2}{n}}}$$

where $\bar{x}_1$ and $\bar{x}_2$ are the means of the two groups, $\hat{s}_1^2$ and $\hat{s}_2^2$ are the sample variances of the two groups, and n is the sample size (the same in each group).

Q5. Complete the function below to calculate this test statistic from the input data:

Hide

```
get_t_stat <- function(data_series1, data_series2) {
  # calculate both means and variances and the sample size

  # calculate the test statistic

  # return the final value
}
```

Click For Answer

A5. Here is an example of the completed function:

Hide

```
get_t_stat <- function(data_series1, data_series2) {
  # calculate both means and variances and the sample size
  m1 <- mean(data_series1)
  m2 <- mean(data_series2)

  v1 <- var(data_series1)
  v2 <- var(data_series2)

  n <- length(data_series1)

  # calculate the test statistic
  ret <- (m1 - m2) / sqrt((v1 + v2) / n)

  # return the final value
  return(ret)
}
```

Q6. Use your completed function to calculate the t-test statistic on the COI data. What value do you get?

Click For Answer

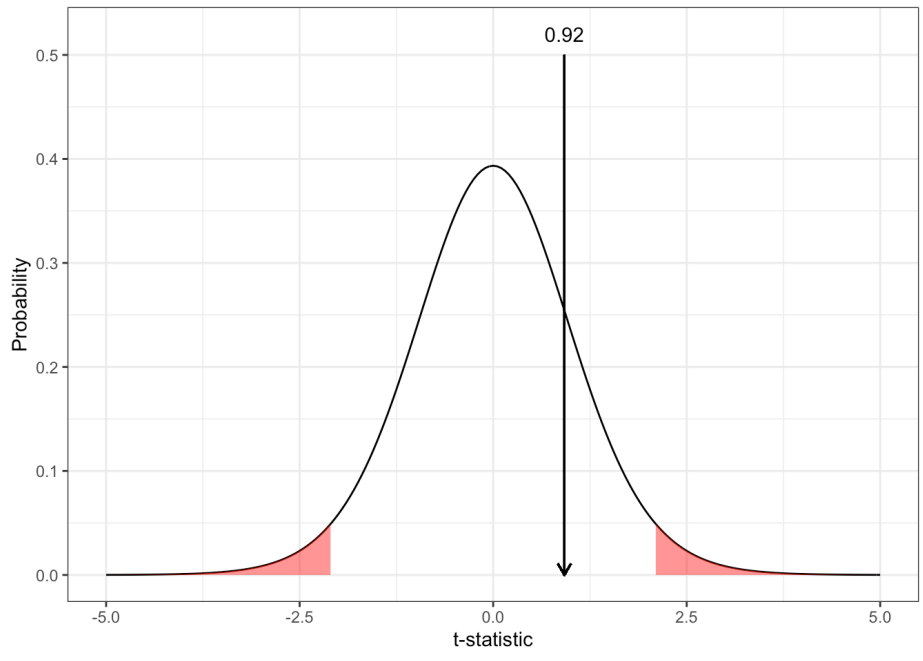
A6. We get the following value:

Hide

```
get_t_stat(COI_control, COI_nets)
```

```
## [1] 0.920358
```

Remember from the lecture that every test statistic has a known distribution under the null hypothesis. In this case, the distribution is called the t-distribution (which makes sense, this is a t-test after all). This distribution has a "degrees of freedom" parameter, which for this test is given by the formula $2n - 2$. For $n = 10$ this gives a value of 18. The following plot shows the t-distribution with 18 degrees of freedom, and with our observed value indicated with an arrow:



Q7. Does your observed value of the test statistic lie in the body of the distribution, or in the tails? What does that tell you about how likely this value is under the null hypothesis of no difference in COI between groups?

Click For Answer

A7. The value lies in the body of the distribution. This means we are fairly likely to see a value as extreme as this if the null hypothesis is true.

We can quantify how extreme this test statistic is using the p-value.

Q8. Complete the code below to calculate a p-value:

Hide

```
# define sample size and get t statistic
n <- # TO COMPLETE
t_stat <- # TO COMPLETE

# calculate p-value
2*pt(abs(t_stat), df = 2*n - 2, lower.tail = FALSE)
```

Click For Answer

A8. Here is an example of the completed code:

Hide

```
# define sample size and get t statistic
n <- 10
t_stat <- get_t_stat(COI_control, COI_nets)

# calculate p-value
2*pt(abs(t_stat), df = 2*n - 2, lower.tail = FALSE)
```

Q9. What is your p-value? Is this significant at the $\alpha = 0.05$ level?

Click For Answer

A9. p-value is around 0.37, which is greater than 0.05 so not significant at the 5% level. Based on this value, we cannot reject the null hypothesis that there is no difference between COI in the control and bednets populations.

There is an easier way of carrying out this type of t-test in R, we can use the `t.test()` function. Run the following code - do you get the same values you calculated by hand?

Hide

```
# Two-sample t-test assuming equal variances between groups
t.test(COI_control, COI_nets, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  COI_control and COI_nets
## t = 0.92036, df = 18, p-value = 0.3696
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5130891  1.3130891
## sample estimates:
## mean of x mean of y
##       2.1       1.7
```

Power analysis and sample size calculation

Before carrying out a study like the one above, it is a good idea to perform a power analysis. This can tell us the chance of finding something interesting if it is really there. More exactly, this tells us the chance of correctly rejecting the null hypothesis given certain assumptions about effect size and sample size.

Above, we used this formula for the t-test statistic:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\hat{s}_1^2 + \hat{s}_2^2}{n}}}$$

Let's rewrite this slightly. First, we will use d to represent the difference between the mean COI in the two populations. It is this difference that we are interested in - as it is the differene between our two study treatments, interventions, *etc.*, and so this is often given the special name **effect size**. A larger effect size means there is a bigger difference between our populations, and so we are more likely to detect it. Second, we will use s^2 to represent the variance in COI in both populations (remember, we have assumed variance is the same in our two populations). The new version of the formula becomes:

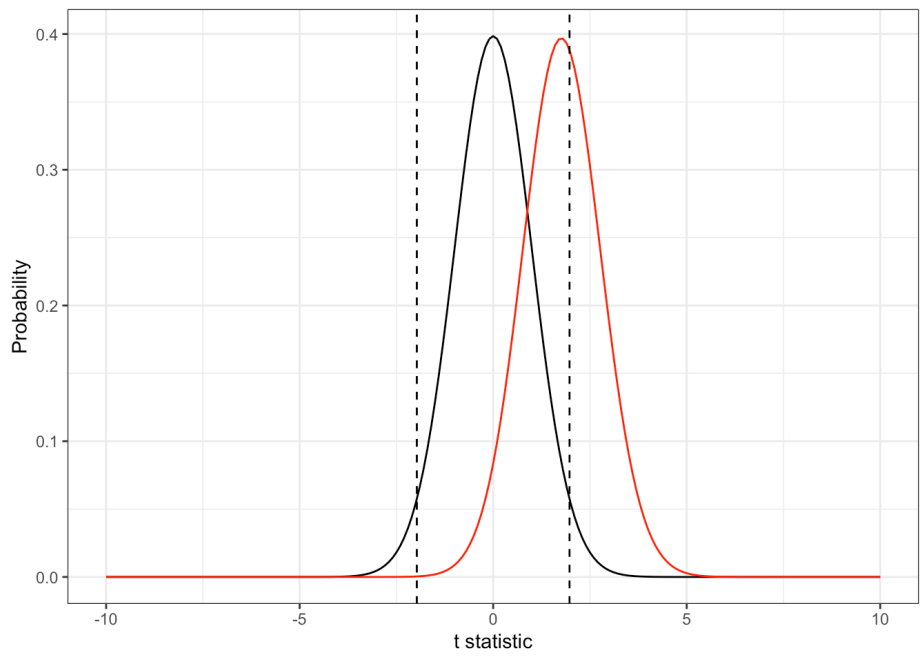
$$t = \frac{d}{\sqrt{\frac{2s^2}{n}}}$$

The following block of code defines these parameters and then plots the distribution of the test statistic under the null and alternative hypotheses. The critical values, or "cut off" points for our designated α value (level of significance), of the null distribution are shown as vertical dashed lines. Recall that the power is the proportion of the red distribution that lies outside of these lines.

Hide

```
# input parameters
d <- 0.5
s <- 2
n <- 100
alpha <- 0.05

# produce plot
plot_ttest(d, s, n, alpha)
```



Q10. Experiment by changing the input parameters in the code above

- What happens when you increase the effect size (`d`)?
- What happens when you increase the standard deviation (`s`)?
- What happens when you increase the sample size (`n`)?
- What happens when you increase the significance threshold (`alpha`)?

Click For Answer

A10.

- Increasing `d` causes the distributions to separate, and so increases power
- Increasing `s` causes the distributions to come together, and so decreases power
- Increasing `n` causes the distributions to separate, and so increases power
- Increasing `alpha` causes the dashed lines to come together, and so increases power

The following function returns the area of the red curve that is outside the dashed lines (i.e. the power). Copy this function into your console:

Hide

```
# returns the power under the t-test
get_pow_ttest <- function(d, s, n, alpha = 0.05) {
  pt(qt(alpha / 2, df = 2*n - 2), df = 2*n - 2, ncp = d / sqrt( 2*s^2 / n)) +
    pt(qt(1 - alpha / 2, df = 2*n - 2), df = 2*n - 2, ncp = d / sqrt( 2*s^2 / n),
        lower.tail = FALSE)
}
```

Q11. Experiment by changing the input parameters in the code below. For `d = 0.5`, `s = 1`, `alpha = 0.05`, can you find a value of `n` that achieves 80% power?

Hide

```
# input parameters
d <- 0.5
s <- 1
n <- 30
alpha <- 0.05

# calculate power
get_pow_ttest(d, s, n, alpha)
```

```
## [1] 0.4778965
```

Click For Answer

A11. A value of `n = 65` is needed to achieve 80% power.

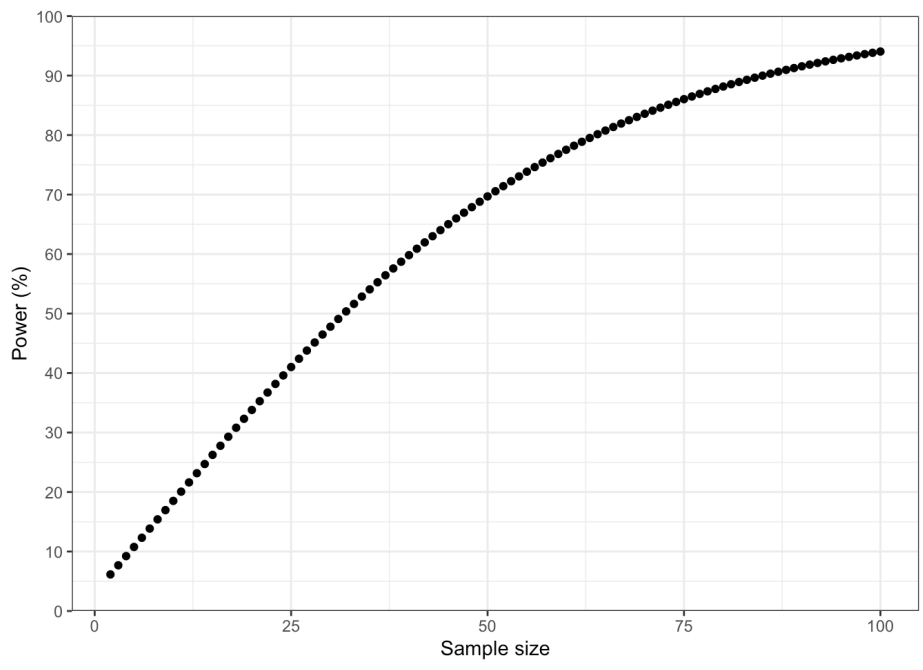
Sometimes it can be useful to look at **power curves**. These show power on the y-axis, and some other variable on the x-axis, usually the sample size.

Copy this code into your console to produce a power curve as a function of `n`:

Hide

```
# input parameters
d <- 0.5
s <- 1
n <- 2:100
alpha <- 0.05

# plot power curve
qplot(x = n, y = get_pow_ttest(d, s, n, alpha)*100) + theme_bw() +
  scale_y_continuous(breaks = seq(0, 100, 10), limits = c(0, 100), expand = c(0,
        0)) +
  xlab("Sample size") + ylab("Power (%)")
```



Q12. Experiment with different values of the input parameters in the code above and see how these change the shape of the curve. What do you notice about the shape of this curve? Does power increase more when we go from `n = 25` to `n = 50`, or from `n = 50` to `n = 75`?

Click For Answer

A12. The curve gets flatter for increasing sample size. There is a greater increase in power going from `n = 25` to `n = 50` compared with going from `n = 50` to `n = 75`. This means we hit *diminishing returns* as we get more and more samples.

Power curves are great for getting an idea how many samples we might want in a perfect world, however, in reality there are other constraints. It might not be logistically feasible to get large sample sizes, or it may be too costly. This does not mean that we should abandon power analysis altogether - instead we should try to work within the constraints. One way of doing this is by fixing the sample size and instead looking at what

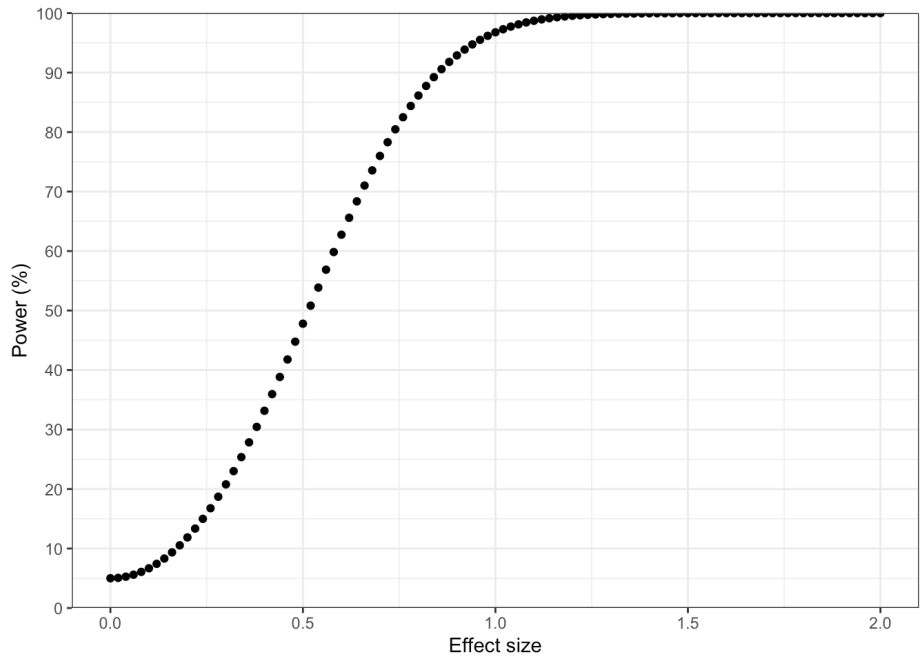
effect size we are powered to detect.

The following code produces a power curve for a fixed sample size, and with the effect size (`d`) on the x-axis. Copy this code into your console and experiment with different values of the input parameters.

Hide

```
# input parameters
d <- seq(0, 2, l = 101)
s <- 1
n <- 30
alpha <- 0.05

# plot power curve
qplot(x = d, y = get_pow_ttest(d, s, n, alpha)*100) + theme_bw() +
  scale_y_continuous(breaks = seq(0, 100, 10), limits = c(0, 100), expand = c(0,
        0)) +
  xlab("Effect size") + ylab("Power (%)")
```



Q13. Imagine that your sample size is fixed by logistical constraints at `n = 17` . Produce power curve under this limitation. What effect size can you detect with 80% power? What does this mean about the COI in the two groups that you are comparing?

Click For Answer

A13. With `n = 17` , we can detect an effect size of about 1.0 with 80% power. This means the COI in the control group needs to be about one higher, on average, than the COI in the bednets group for us to have a good chance of detecting a difference.

From the analysis above, we can see that the orginal study, which used a sample size of `n = 10` , had extremely low power. We were only powered to detect a significant difference between groups if there was a difference in average COI of more than 1.5, which is quite a large difference. On the other hand, if we wanted to detect differences of COI down to 0.5 then we would need a sample size closer to `n = 65` . Given these findings, it is not surprising that we got a non-significant result in the real data analysis. Unfortunately, this study was probably a waste of both time and money. It may have generated some interesting descriptive statistics, and the results can be used as pilot data when designing future studies, but in terms of answering the key scientific question it was doomed to failure from the outset. ***The hard truth is - not all studies are worth doing!***

Comparing DR mutation prevalence

Statistical testing

You have been asked by your National Malaria Control Programme (NMCP) to establish whether mutations at the Chloroquine resistance locus *pfcrt* have increased significantly in your study area between 2005 to 2020. Two cross-sectional surveys have been conducted, with a large number of samples obtained and successfully sequenced in each year.

Let’s load data and take a look:

Hide

```
# load COI data
load("data/pfcrt.RData")

head(pfcrt)
```

```
##   year       ID pfcrt
## 1 2005 ID2005.1     0
## 2 2005 ID2005.2     0
## 3 2005 ID2005.3     0
## 4 2005 ID2005.4     1
## 5 2005 ID2005.5     1
## 6 2005 ID2005.6     0
```

We can see that our data is arranged in a data.frame, and in *long format* meaning all factors (e.g. year) are in columns. For each individual ID we have a binary 1/0 value for whether the sample contained the *pfcrt* mutation. Our first task is to summarise this data to tell us:

- The number of samples obtained in each year
- The proportion of samples containing the *pfcrt* mutation (i.e. the prevalence of the mutation)

Q14. Complete the following code to 1) group the data by year, and 2) summarise to obtain the prevalence of *pfcrt* mutations. Has the prevalence of the mutation increased or decreased over time?

Hide

```
pfcrt_summary <- pfcrt %>%
  group_by( # TO COMPLETE ) %>%
  summarise(n = n(),
            prev = #TO COMPLETE )

pfcrt_summary
```

Click For Answer

A14. Here is an example of the completed code. The mutation has more than doubled in prevalence over time.

Hide

```
pfcrt_summary <- pfcrt %>%
  group_by(year) %>%
  summarise(n = n(),
            prev = mean(pfcrt))

pfcrt_summary
```

```
## # A tibble: 2 × 3
##    year     n  prev
##   <dbl> <int> <dbl>
## 1  2005   800 0.29
## 2  2020  1200 0.615
```

For our statistical test, we want to compare two values as in the COI example above, but this time our values are proportions, and so are constrained to be between 0 and 1. The appropriate test here is not the t-test, but rather the two-proportion Z-test. The test statistic is calculated as follows:

$$Z = \frac{\hat{p}_1 - \hat{p}_2}{\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}}}$$

where $\hat{p}_1$ and $\hat{p}_2$ are the prevalences in the two groups and n_1 and n_2 are the sample sizes. $\bar{p}$ is the mean prevalence, calculated as $\bar{p} = \frac{n_1\hat{p}_1 + n_2\hat{p}_2}{n_1 + n_2}$ The following code calculates this statistic:

Hide

```
# get values from summary table
n <- pfcrt_summary$n
p <- pfcrt_summary$prev
p_bar <- sum(n*p) / sum(n)

# calculate Z statistic
Z <- (p[1] - p[2]) / sqrt(p_bar*(1 - p_bar) * (1/n[1] + 1/n[2]))
Z
```

```
## [1] -14.2472
```

For the z-test, the distribution of the test statistic under the null hypothesis is the z-distribution, also called the **normal distribution**. For the normal distribution, the 5% "tails" are at -1.96 and +1.96.

Q15. Is your observed Z value in the body or the tails of the distribution? What does this tell you about how likely we are to see a value this extreme by chance?

Click For Answer

A15. The observed value is a long way into the tails of the distribution. We are extremely unlikely to see a value as extreme as this under the null hypothesis of no difference in prevalence between groups.

We can calculate a p-value as follows:

Hide

```
# calculate p-value
2*pnorm(abs(Z), lower.tail = FALSE)
```

```
## [1] 4.666238e-46
```

Q16. Is the difference in *pfcrt* prevalence significant at the 5% threshold? Would you accept or reject the null hypothesis?

Click For Answer

A16. The p-value is very small, far below the 5% threshold. In this case we would reject the null hypothesis of no difference between groups. In other words, we conclude that there is a large difference (in this case an increase) in *pfcrt* prevalence from 2005 to 2020.

As with the t-test, there is an easier way of carrying out a two-proportion Z-test in R, we can use the `prop.test()` function. This carries out the same analysis we just did by hand above:

Hide

```
prop.test(x = n*p, n = n, correct = FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  n * p out of n
## X-squared = 202.98, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.3667931 -0.2832069
## sample estimates:
## prop 1 prop 2
##  0.290  0.615
```

Power analysis and sample size calculation

As before, we will conduct a power analysis to work out if this study was well designed. We will keep things simple by assuming the same number of samples in both years. We can rewrite the formula for the Z statistic as follows:

$$Z = \frac{p_1 - p_2}{\sqrt{\frac{\bar{2p}(\bar{1-p})}{n}}}$$

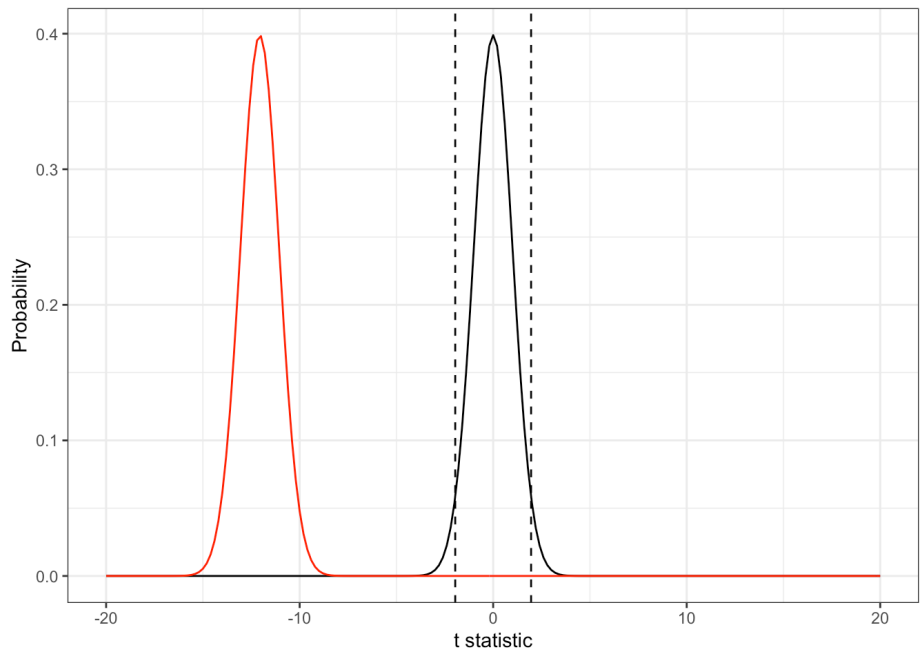
where n is now the sample size in both years (replacing n_1 and n_2 in the previous version).

The following code plots the distribution of the Z statistic under the null and alternative hypotheses, similar to what we did for the t-test example above. Experiment with different values and see how they affect the distributions.

Hide

```
# variable parameters
p1 <- 0.3
p2 <- 0.6
n <- 800
alpha <- 0.05

# produce plot
plot_ztest(p1, p2, n, alpha)
```



We can also write a function to calculate the power (the area of the red distribution that is beyond the dashed lines) exactly:

Hide

```
# function that returns the power given these parameters
get_pow_ztest <- function(p1, p2, n, alpha = 0.05) {
  p_bar <- mean(c(p1, p2))
  alt_mean <- (p1 - p2) / sqrt(2*p_bar*(1 - p_bar) / n)
  pnorm(qnorm(alpha / 2), mean = alt_mean) + pnorm(qnorm(1 - alpha / 2), mean = al
        t_mean, lower.tail = FALSE)
}
```

Q17. When calculating power, we need to make assumptions about the effect size (in this case the true prevalence in both years) and the sample size. The NMCP have asked you to calculate power under the following assumptions:

- Prevalence doubles from 40% to 80%
- Prevalence doubles from 30% to 60%
- Prevalence doubles from 20% to 40%
- Prevalence increases from 30% to 45%

Assume a sample size of `n = 1000` throughout.

Click For Answer

A17. Power is very high, more than 99.99% in all cases.

Hide

```
get_pow_ztest(p1 = 0.4, p2 = 0.8, n = 1000)
get_pow_ztest(p1 = 0.3, p2 = 0.6, n = 1000)
get_pow_ztest(p1 = 0.2, p2 = 0.4, n = 1000)
get_pow_ztest(p1 = 0.3, p2 = 0.45, n = 1000)
```

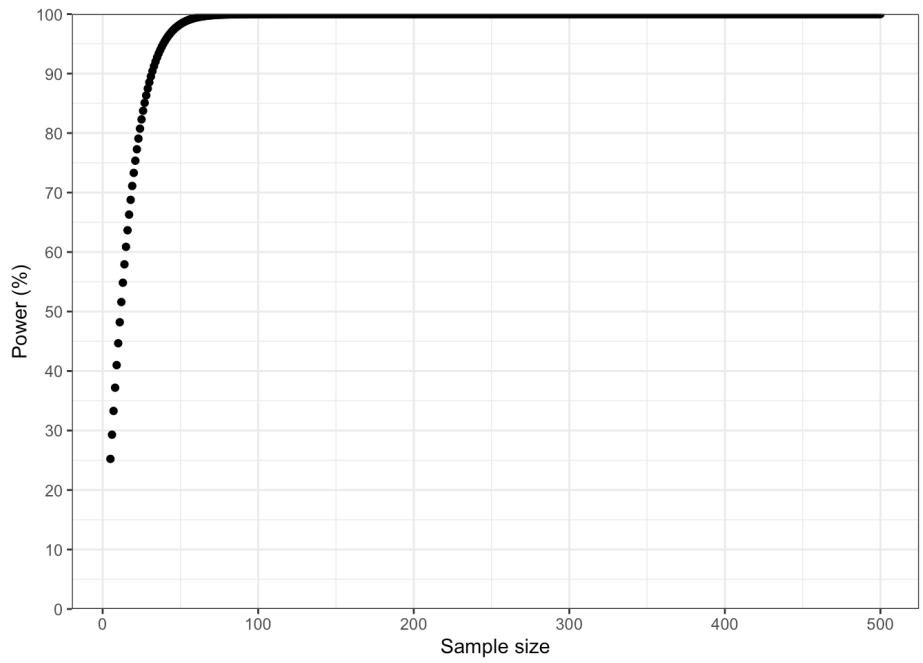
```
## [1] 1
## [1] 1
## [1] 1
## [1] 0.9999997
```

We can produce a power curve by passing in a range of values of `n` into our function. Experiment with different parameters in the code below to see how they change the shape of the power curve:

Hide

```
# variable parameters
p1 <- 0.4
p2 <- 0.8
n <- 5:500

# plot power curve
qplot(x = n, y = get_pow_ztest(p1, p2, n, alpha)*100) + theme_bw() +
  scale_y_continuous(breaks = seq(0, 100, 10), limits = c(0, 100), expand = c(0,
        0)) +
  xlab("Sample size") + ylab("Power (%)")
```



Q18. Produce power curves for the four scenarios you have been asked to explore by the NMCP. Approximately what sample size is needed in each case to achieve a power of 80%?

Click For Answer

A18. Approximate sample sizes needed are 20, 40, 80 and 190 for the four scenarios.

Sometimes it is possible to come up a formula for the sample size. In this case, the formula can be written:

$$n = (z_{\alpha/2} + z_{\beta})^2 \frac{2\bar{p}(1-\bar{p})}{(p_1-p_2)^2}$$

where $z_{\alpha/2}$ is the critical value at a significance level α (two-tailed), which we have already noted is around 1.96. z_{β} is a similar value, this time calculated from β which is defined as 1 minus the desired power (in our case $\beta = 0.2$ for 80% power). The following function implements this formula to give the sample size needed for any given power:

Hide

```
get_n_ztest <- function(p1, p2, alpha = 0.05, power = 0.8) {
  p_bar <- mean(c(p1, p2))
  (qnorm(1 - alpha / 2) + qnorm(power))^2 * 2*p_bar*(1 - p_bar) / (p1 - p2)^2
}
```

This formula will sometimes give non-integer values, in which case they should be rounded up to the nearest whole number.

Q19. Use this exact function to calculate the sample size required for each of the four scenarios requested by the NMCP. What values do you get? Which of these values should you take forward as your final recommendation?

Click For Answer

A19. For the scenarios below, we get sample sizes of 24, 44, 83 and 187. We should take the largest of these forwards as our recommendation, as then we will have sufficient power for any of the scenarios.

Hide

```
get_n_ztest(p1 = 0.4, p2 = 0.8, power = 0.8)
get_n_ztest(p1 = 0.3, p2 = 0.6, power = 0.8)
get_n_ztest(p1 = 0.2, p2 = 0.4, power = 0.8)
get_n_ztest(p1 = 0.3, p2 = 0.44, power = 0.8)
```

```
## [1] 23.54664
## [1] 43.16884
## [1] 82.41324
## [1] 186.6912
```

Q20. The cost of the study, taking into account clinical time, lab time and the cost of sequencing, is estimated at 50 USD per sample. What is the estimated cost of the original study that used 800 samples in 2005 and 1200 samples in 2020? What is the cost of your new study design based on your power analysis (remember that the `n` you have calculated is for *each* of the two years)?

Click For Answer

A20. The cost of the original study was 100,000 USD. With a sample size of 187 per year, the new cost goes down to 18,700 USD.

What we have seen here is an example of an over-powered study. While we may think there is no harm in collecting more samples, we should keep in mind that every study has costs and that funds might be better spent elsewhere. In this case, we found that a properly powered design was more than 5 times cheaper than the original study. The NMCP could have conducted the same study in 5 different parts of the country, or over multiple years, and still would have made a cost saving. So, while it's generally a good idea to be cautious when calculating sample sizes, opting for larger values where unsure, there are limits to this approach and it is possible to collect too many samples. Whatever the situation, conducting a formal power analysis **before** the study has been carried out is an excellent way of exploring these issues.

One final thing to note is that the formulae used in this power calculation are only approximations. Generally, simulation or computational-based methods will provide more accurate results as they don't make these same approximations. The `get_pow_ztest_exact()` function below uses this approach to calculate power exactly, which can be compared against the `get_pow_ztest()` function used above. Can you find parameter combinations where they agree/disagree?

Hide

```
get_pow_ztest(p1 = 0.2, p2 = 0.5, n = 20)
get_pow_ztest_exact(p1 = 0.2, p2 = 0.5, n = 20)
```

```
## [1] 0.5116136
## [1] 0.5308455
```

Dealing with dropout

One thing that we need to be aware of when calculating sample sizes is dropout. This to refers to anything that causes our final sample size to be less than what we originally planned. Dropout can occur for many reasons, including:

- People withdrawing consent from the study
- People dying or migrating out of the study area
- Samples not meeting the criteria required for analysis (for example, being non-*vivax* in a *falciparum* study)
- Samples being lost or contaminated
- Samples failing sequencing, for example due to low parasitaemia

We need to account for dropout in our sample size calculations to ensure that we have enough samples left over for our final analysis. The formula for adjusting for dropout is fairly simple:

$$n_{\text{adjusted}} = \frac{n_{\text{original}}}{1-p_{\text{dropout}}}$$

where n_{original} is the raw sample size that we get out of our power analysis, and p_{dropout} is the proportion of dropout that expect for a particular reason. For example, if our original sample size is `n = 100` and we expect 20% dropout then we do $n_{\text{adjusted}} = 100/(1-0.2) = 100/0.8 = 125$. So, we need 125 people to account for this much dropout. We can easily check this: if we have 125 people and 20% of them drop out then we lose 25 people, bringing us back down to 100.

Q21. A control programme wants to determine whether the frequency of *dhps* K540E mutations in their country is above 10%. If so, they plan on switching first line drugs away from Sulfadoxine-Pyrimethamine. They plan on replicating this study in 5 distinct regions throughout the country. A statistical sample size calculation has found that 220 samples will be needed to achieve the power they want. However, each of the 5 laboratories involved in processing the samples has different levels of experience, resulting in different rates of samples being lost. The estimated proportion of samples lost in each of the labs is as follows:

Lab1: 10% Lab2: 3% Lab3: 14% lab4: 25% lab5: 9%

What adjusted sample size is needed for each of the 5 regions? Remember to round values up to the nearest whole number.

Click For Answer

A21. Adjusted sample sizes can be calculated as follows. The `ceiling()` function ensures that values are rounded up.

Hide

```
# define proportion dropout in each of the labs
p_dropout <- c(lab1 = 0.1, lab2 = 0.03, lab3 = 0.14, lab4 = 0.25, lab5 = 0.09)

# calculate adjusted sample sizes
ceiling(220 / (1 - p_dropout))
```

```
## lab1 lab2 lab3 lab4 lab5
##  245  227  256  294  242
```

Sometimes, we need to perform the adjustment above multiple times. For example, we might expect to lose 5% of samples due to withdrawing consent, and of the samples remaining we expect to lose 10% due to sequencing failure. In this case we should first adjust for the 5% loss and then *using the new adjusted value* we should adjust for the 10% loss. Note that this does not give exactly the same as if we account for the full 15% in one go.

Q22. A control programme is running a study in which they follow people over a period of 6 months and measure incidence of malaria. Parasites will be genotyped periodically to determine if the same or different genotypes are present. Statistical sample size calculation has indicated that they need 400 samples in total over the 6 month period. They expect to lose 15% of samples due to loss-to-followup (e.g. people migrating out or dropping out of the study). Of those who are sequenced, they expect 10% of samples to fail. What is the final adjusted sample size they need?

Click For Answer

A22. Adjusted sample sizes can be calculated as follows.

Hide

```
ceiling(400 / (1 - 0.15) / (1 - 0.1))
```

```
## [1] 523
```

Putting it into practice

Hopefully by this point you feel comfortable with the basics of power analysis and sample size calculation. The examples above were designed to illustrate key learning points, but real-world analyses tend to be a bit messier and involve some creative thinking. Have a go at approaching the following more realistic problem.

Q23. (long excercise) You have been recruited by the NMCP of Zambia to conduct a study into the changing prevalence of *dhps* K540E mutations. They have a series of samples that were collected in a pilot study in 2001 from 5 different sampling locations. These samples have been sequenced, and give baseline estimates of the prevalence of K540E mutations in each of the locations. They plan to conduct a present-day study in the same locations to determine whether the prevalence has changed significantly over this time period.

Here is the pilot data:

Hide

```
load("data/Zambia_pilot.RData")
Zambia_pilot
```

```
##   location total_samples K540E
## 1    Kabwe            80    11
## 2    Ndola            24     6
## 3  Chipata           110    13
## 4    Mansa            90    14
## 5   Lusaka            70    12
```

You also have some information on logistical constraints. Samples will be sequenced in several different laboratories, and you have estimates of the fraction expected to fail in each location:

Hide

```
load("data/Zambia_logistics.RData")
Zambia_logistics
```

```
##   location fail_fraction
## 1    Kabwe          0.10
## 2    Ndola          0.06
## 3  Chipata          0.30
## 4    Mansa          0.02
## 5   Lusaka          0.04
```

The budget of the study allows for 1000 samples to be sequenced in total over all 5 locations.

In this example we know the sample size in the first group (n_1) and we are trying to work out the sample size required in the second group (n_2). This leads to the following formula:

$$n_2 = \frac{p_2(1-p_2)}{\frac{(p_1-p_2)^2}{(z_{\alpha/2}+z_\beta)^2} - \frac{p_1(1-p_1)}{n_1}}$$

This formula only holds as long as $n_1 > (z_{\alpha/2} + z_\beta)^2 \frac{p_1(1-p_1)}{(p_1-p_2)^2}$. If n_1 is smaller than this value then it is impossible to reach the desired power with any sample size.

You have been asked to:

1. Estimate the prevalence of K540E mutations in 2001 from the pilot data.
2. Write a new function to implement the sample size formula above. You might find it useful to look at the `get_n_ztest()` function defined in the previous as a guide.
3. Use your new function to perform sample size calculation in each location, assuming prevalence has doubled since the pilot study. Aim for 80% power.
4. Adjust sample sizes to account for dropout.
5. Calculate your total number of samples for the study. Is this within budget?
6. If not, is there a location that you could drop to bring it within budget? Justify your choice of location.
7. Produce a summary paragraph to send to the NMCP with your recommendation. This should outline your assumptions as well as your findings. It should contain a clear value for the sample size required in each location.

Click For Answer

A23. We can start by calculating the prevalence from the pilot data:

Hide

```
Zambia_analysis <- Zambia_pilot %>%
  mutate(prev_2001 = K540E / total_samples)

Zambia_analysis
```

```
##   location total_samples K540E prev_2001
## 1    Kabwe            80    11 0.1375000
## 2    Ndola            24     6 0.2500000
## 3  Chipata           110    13 0.1181818
## 4    Mansa            90    14 0.1555556
## 5   Lusaka            70    12 0.1714286
```

Next, we need to define a function to implement the new sample size formula. Here is an example of what this might look like. Note, this function contains a check to ensure that a warning is produced if no finite sample size is possible.

Hide

```
  part3 <- p1*(1 - p1) / n1

  # get final value
  ret <- part1 / (part2 - part3)

  # replace with NA if outside range and throw warning
  ret[part2 < part3] <- NA

  # check that there is a value of n2 that is valid
  if (any(is.na(ret))) {
    warning("There is no finite value of n2 that achieves the desired power")
  }

  return(ret)
}
```

We have been asked to assume that prevalence doubles in each location:

Hide

```
Zambia_analysis <- Zambia_analysis %>%
  mutate(prev_now = prev_2001 * 2)

Zambia_analysis
```

```
##   location total_samples K540E prev_2001  prev_now
## 1    Kabwe            80    11 0.1375000 0.2750000
## 2    Ndola            24     6 0.2500000 0.5000000
## 3  Chipata           110    13 0.1181818 0.2363636
## 4    Mansa            90    14 0.1555556 0.3111111
## 5   Lusaka            70    12 0.1714286 0.3428571
```

We can use our `get_n2_ztest()` function to calculate the optimal sample size in each location, assuming 80% power:

Hide

```
Zambia_analysis <- Zambia_analysis %>%
  mutate(n_raw = get_n2_ztest(prev_2001, prev_now, total_samples, alpha = 0.05, po
        wer = 0.8))

Zambia_analysis
```

```
##   location total_samples K540E prev_2001  prev_now     n_raw
## 1    Kabwe            80    11 0.1375000 0.2750000  215.2238
## 2    Ndola            24     6 0.2500000 0.5000000 1662.0150
## 3  Chipata           110    13 0.1181818 0.2363636  216.9228
## 4    Mansa            90    14 0.1555556 0.3111111  132.0203
## 5   Lusaka            70    12 0.1714286 0.3428571  131.3705
```

We need to adjust these raw values to account for dropout. We can do this by merging with the logistics data.frame and then using the dropout formula:

Hide

```
Zambia_analysis <- Zambia_analysis %>%
  left_join(Zambia_logistics) %>%
  mutate(n_adjusted = ceiling(n_raw / (1 - fail_fraction)))

Zambia_analysis
```

```
##   location total_samples K540E prev_2001  prev_now     n_raw fail_fraction
## 1    Kabwe            80    11 0.1375000 0.2750000  215.2238          0.10
## 2    Ndola            24     6 0.2500000 0.5000000 1662.0150          0.06
## 3  Chipata           110    13 0.1181818 0.2363636  216.9228          0.30
## 4    Mansa            90    14 0.1555556 0.3111111  132.0203          0.02
## 5   Lusaka            70    12 0.1714286 0.3428571  131.3705          0.04
##   n_adjusted
## 1        240
## 2       1769
## 3        310
## 4        135
## 5        137
```

Finally, we can calculate the total sample size of the study:

Hide

```
sum(Zambia_analysis$n_adjusted)
```

```
## [1] 2591
```

We find that 2591 samples are needed in total, which is beyond our budget of 1000 samples. Looking at the locations, the vast majority of these samples are from the Ndola region. The reason for the huge sample size in this region is that we had very little pilot data - just 24 samples. With so few samples we have little confidence in what the true prevalence was in 2001, and therefore we need a large number of samples to conclusively say that prevalence has doubled. If we exclude just this one region then we end up within budget. We may want to collect some samples from Ndola for other reasons, for example to give us a new baseline for subsequent studies, but if our goal is to detect changes in prevalence over time then this region is unlikely to yield significant results.

In summary, this is what we would tell the NMCP:

We recommend to collect the following sample sizes:

Hide

```
Zambia_analysis %>%
  filter(location != "Ndola") %>%
  select(location, n_adjusted)
```

```
##   location n_adjusted
## 1    Kabwe        240
## 2  Chipata        310
## 3    Mansa        135
## 4   Lusaka        137
```

These sample sizes give us 80% power in each location to detect a doubling in K540E prevalence since 2001. Sample sizes have been adjusted to account for the fraction expected to fail in laboratory procedures. The Ndola region has been excluded from this analysis because pilot data from this area was very weak, meaning we were unable to achieve the desired power within budget constraints.

- Dependencies for Practical
- Intro to statistical testing and power analysis
- Comparing COI between two populations
  - Statistical testing
  - Power analysis and sample size calculation
- Comparing DR mutation prevalence
  - Statistical testing
  - Power analysis and sample size calculation
- Dealing with dropout
- Putting it into practice