Code

AMMS Practical: Power Analysis for Clustered Designs

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/power2_utils.R")
```

Intro to clustered surveys

There are many situations where it makes sense to sample in clusters, rather than sampling individuals. Some reasons include:

- It is logistically infeasible to sample at the individual level. This is true in province- or country-level surveys, where we cannot sample individuals from the entire population at random as they could be located anywhere, which would be a logistical nightmare!
- The population naturally groups itself into clusters. An example would be sampling malaria in clinics, which concentrate the malaria cases of the entire catchment area, making surveillance at this level highly efficient.
- Interventions are planned at the cluster level, rather than the individual level. If an intervention will be delivered in a clustered way then it may make sense to collect baseline data in a similar fashion.

When using a clustered design, we need to be aware that observations will tend to be more similar within clusters than between clusters - referred to as **within-cluster correlation**, or **intra-cluster correlation**. For example, as a general rule individuals from the same village will tend to be more similar to each other in behaviours, physical characteristics and risk factors than individuals sampled at random from the wider province. In infectious diseases we have another reason for within-cluster correlation; disease transmission. This can lead to local outbreaks that cause a large number of correlated outcomes in a single cluster.

Fortunately, there are well-defined statistical methods for dealing with intra-cluster correlation. Using these methods we can estimate the stregnth of this correlation, and then account for it in our statistical testing.

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 of cluster surveys, 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:

- Identify overdispersed data
- Estimate an intra-cluster correlation coefficient (ICC) and a design effect
- Construct a confidence interval that takes account of intra-cluster correlation
- Test for a difference between a cluster-sampled prevalence and an arbitrary threshold (for example in *pfhrp2/3* study design)

- Perform power analysis and sample size calculation for clustered surveys

Increased uncertainty due to clustering

Intra-cluster correlation and overdispersion

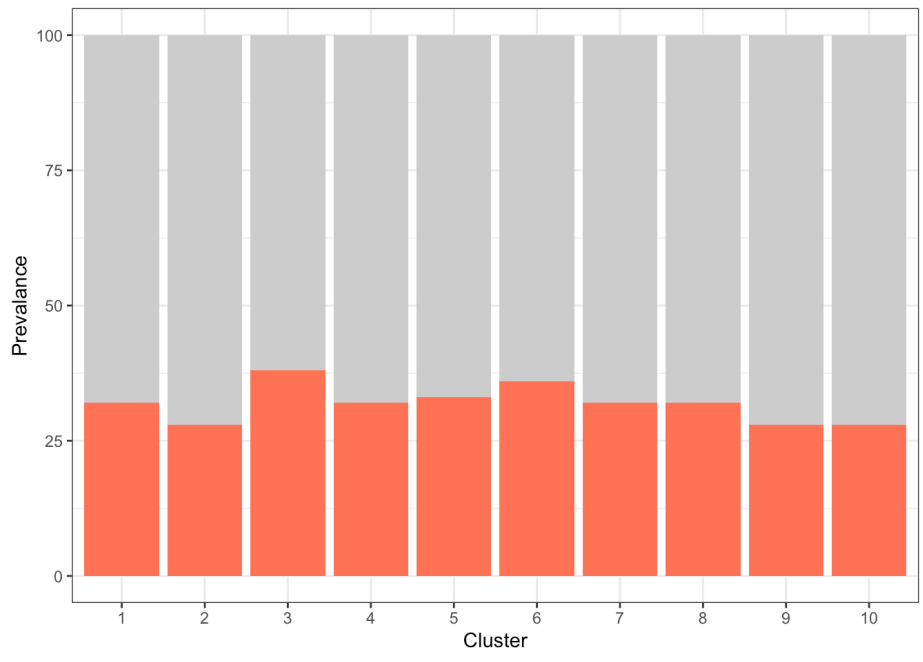
Before getting into ideas around intra-cluster correlation, it is worth exploring what we would expect to see if there was *no* within-cluster correlation. The following code simulates the observed prevalence in a series of clusters, where every individual in every cluster has the same probability `prev` of being positive for malaria. It then produces a simple barplot of observed prevalence.

Hide

```
# set simulation parameters
n_clusters <- 10    # number of clusters
n_samp <- 100       # number of samples per cluster
prev <- 0.3         # true prevalence

# simulate some data by drawing from binomial distribution
cluster_prev <- data.frame(cluster = 1:n_clusters,
                           p = rbinom(n_clusters, n_samp, prev) / n_samp)

# produce a simple barplot of prevalence
barplot_clusters(cluster_prev)
```



Q1. Try running this simulation code multiple times to get a feel for the distribution of prevalence between clusters. Do you ever see a cluster with prevalence of 0%? Do you ever see a cluster with prevalence above 50%?

Click For Answer

In this situation, every individual has the same probability of being positive irrespective of what cluster they are in. This means we can pool results over clusters without losing anything. In the example above we have the same number of samples in each cluster, meaning the total number of samples is given by `n_clusters * n_samp`. We know from the previous power analysis practical how to use this value to construct a 95% confidence interval (CI) on our estimate of the prevalence. The following code performs the same simulation, but now constructs a 95% CI rather than plotting results. Have a look through the mathematics and make sure you are familiar with how this interval is constructed.

Hide

```
# set simulation parameters
n_clusters <- 10    # number of clusters
n_samp <- 100       # number of samples per cluster
prev <- 0.3         # true prevalence

# simulate some data by drawing from binomial distribution
cluster_prev <- data.frame(cluster = 1:n_clusters,
                           p = rbinom(n_clusters, n_samp, prev) / n_samp)

# estimate prevalence as the mean over clusters
p_bar <- mean(cluster_prev$p)

# calculate standard error using n_samp*n_clusters as our total sample size
SE <- sqrt(p_bar*(1 - p_bar) / (n_samp*n_clusters - 1))

# construct a 95% confidence interval
p_bar + c(-1.96, 1.96)*SE
```

```
## [1] 0.2599192 0.3160808
```

Q2. Try running this simulation code multiple times. How often does the true prevalence (`prev`) lie within your 95% CI?

Click For Answer

So, if there is no within-cluster correlation and all individuals have the same probability of being positive irrespective of their cluster, then our data analysis is fairly simple. To estimate prevalence over all clusters (e.g. over the entire geographic region) we first pool results, then we estimate prevalence as the proportion of positive cases. Our 95% CI can be constructed using the standard formula above.

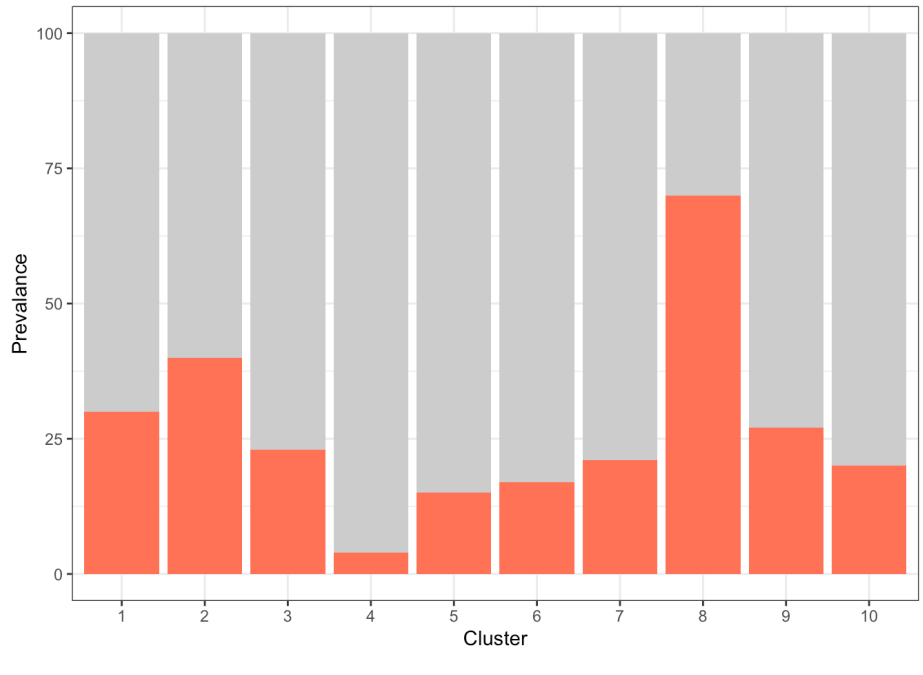
But what if our data are **overdispersed**, meaning the spread is greater than we would expect if probabilities were the same everywhere? The following code simulates data from an overdispersed distribution. The amount of overdispersion is controlled by the intra-cluster correlation coefficient (ICC) which varies between 0 and 1. The larger the ICC, the bigger the spread.

Hide

```
# set simulation parameters
n_clusters <- 10
n_samp <- 100
prev <- 0.3
ICC <- 0.2

# simulate some data from overdispersed distribution
cluster_prev <- data.frame(cluster = 1:n_clusters,
                           p = draw_overdispersed(n_clusters, n_samp, prev, ICC) /
        n_samp)

# produce a simple barplot of prevalence
barplot_clusters(cluster_prev)
```



Q3. As before, try running this simulation code multiple times to get a feel for the distribution of prevalence between clusters. Now do you ever see a cluster with prevalence of 0%? Do you ever see a cluster with prevalence above 50%? What happens when you use a larger ICC?

Click For Answer

In the above we have seen the connection between *overdispersion* and *intra-cluster correlation*. If there is any intra-cluter correlation then this will cause observations to be overdispersed. Likewise, if there is any overdispersion then it must be that there is some intra-cluster correlation. The two ideas are tightly linked, and we will talk about them interchangeably.

If we didn't know about the issues of cluster sampling then we may be tempted to pool these results together and estimate prevalence exactly as we did in the previous example. This is implemented in the following code:

Hide

```
# set simulation parameters
n_clusters <- 10
n_samp <- 100
prev <- 0.3
ICC <- 0.5

# simulate some data from overdispersed distribution
cluster_prev <- data.frame(cluster = 1:n_clusters,
                           p = draw_overdispersed(n_clusters, n_samp, prev, ICC) /
        n_samp)

# estimate prevalence as the mean over clusters
p_bar <- mean(cluster_prev$p)

# calculate standard error using n_samp*n_clusters as our total sample size
SE <- sqrt(p_bar*(1 - p_bar) / (n_samp*n_clusters - 1))

# construct a 95% confidence interval
p_bar + c(-1.96, 1.96)*SE
```

```
## [1] 0.291073 0.348927
```

Q4. Try running this simulation code multiple times. How often does the true prevalence lie within your 95% CI?

Click For Answer

So, for overdispersed data we can get misleading results if we do not take clustering into account. But what if we approach this problem in a completely different way? We could ignore the fact that we have sampe sizes for each of our prevalence estimates, and instead treat them as just distinct values. We can then calculate standard error and a 95% CI as follows:

Hide

```
# set simulation parameters
n_clusters <- 10
n_samp <- 100
prev <- 0.3
ICC <- 0.5

# simulate some data from overdispersed distribution
cluster_prev <- data.frame(cluster = 1:n_clusters,
                           p = draw_overdispersed(n_clusters, n_samp, prev, ICC) /
        n_samp)

# estimate prevalence as the mean over clusters
p_bar <- mean(cluster_prev$p)

# calculate standard error from variance over clusters
SE <- sqrt(var(cluster_prev$p) / (n_clusters - 1))

# construct a 95% confidence interval
p_bar + c(-1.96, 1.96)*SE
```

```
## [1] 0.05798201 0.52001799
```

Q5. Try running this simulation code multiple times. How often does the true prevalence lie within your 95% CI?

Click For Answer

So, one simple way to deal with intra-cluster correlation is to abandon the idea of pooling samples over clusters, and instead treat each cluster as a single observation. We can then calculate a 95% CI for the study-level prevalence based on this much smaller number of values. The CI produced in this way is robust to overdispersion, and does not risk giving false confidence in our conclusions, but it does come at the cost of reduced power. We will explore this below.

Design effects

One way of quantifying the effect of clustering on our analysis is through the **design effect**. This measures how much greater the variance in our estimate of the prevalence is than we would expect under simple random sampling (SRS). In other words, it quantifies the extent to which the design of our study is influencing our precision, hence the name *design effect*. The larger the design effect, the further we are from what we would expect under SRS (a value of 1 indicates that there is no difference from SRS).

For clustered surveys, the design effect can be defined as:

$$D_{\text{eff}} = \frac{\text{Var}_{\text{clust}}(\bar{p})}{\text{Var}_{\text{SRS}}(\bar{p})}$$

where $\bar{p}$ is our estimate of the overall prevalence, $\text{Var}_{\text{clust}}(\bar{p})$ is the actual variance of this estimator taking into account clustering, and $\text{Var}_{\text{SRS}}(\bar{p})$ is the variance that we would expect under simple random sampling. These two variances are simply the square of the two different types of standard error calculated above - one at the cluster level, and one calculated by pooling results. We can write:

$$\text{Var}_{\text{clust}}(\bar{p}) = \frac{s_c^2}{c-1}$$

where s_c^2 is the variance over clusters and c is the number of clusters. For the SRS case, we can write:

$$\text{Var}_{\text{SRS}}(\bar{p}) = \frac{\bar{p}(1-\bar{p})}{nc-1}$$

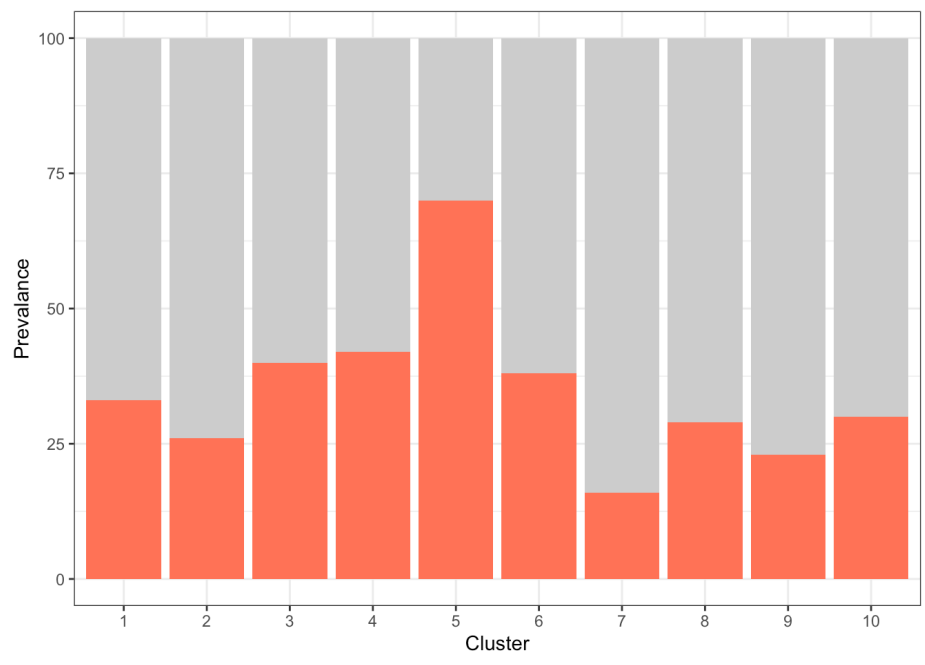
where n is the number of samples per cluster. Once we have these two values we can combine them to produce an estimate of the design effect.

Let's load some data to work on:

Hide

```
# load overdispersed data from file
load("data/prev_overdispersed.RData")

# produce a simple barplot of prevalence
barplot_clusters(prev_overdispersed)
```



Hide

```
prev_overdispersed
```

```
##    cluster n_samp    p
## 1        1    100 0.33
## 2        2    100 0.26
## 3        3    100 0.40
## 4        4    100 0.42
## 5        5    100 0.70
## 6        6    100 0.38
## 7        7    100 0.16
## 8        8    100 0.29
## 9        9    100 0.23
## 10      10    100 0.30
```

Here we have data from 10 clusters, along with the same sample size in each cluster. Looking at the barplot, and using our intuition about what data looks like under independent probabilities, we may already suspect that there is some overdispersion here.

First, we calculate the cluster-level variance:

Hide

```
# calculate actual variance at cluster level
n_clusters <- nrow(prev_overdispersed)
var_clust <- var(prev_overdispersed$p) / (n_clusters - 1)
var_clust
```

```
## [1] 0.002417407
```

Second, we calculate the variance that we would expect under SRS:

Hide

```
# calculate variance we would expect under SRS
p_bar <- mean(prev_overdispersed$p)
n_samp_total <- sum(prev_overdispersed$n_samp)
var_srs <- p_bar*(1 - p_bar) / (n_samp_total - 1)
var_srs
```

```
## [1] 0.0002268178
```

Finally, we can estimate the design effect as one variance over the other.

Q6. Complete the following code to estimate the design effect:

Hide

```
# estimate design effect
Deff <- # TO COMPLETE
Deff
```

Click For Answer

The design effect is just greater than 10. This tells us that the variance of our estimate of the prevalence is 10 times higher than we would expect under simple random sampling.

One useful way of thinking about this result is in terms of the *effective sample size*. This is the sample size that we *would* need to generate data like ours if SRS was indeed true. The effective sample size, n_{eff} is obtained by dividing the actual sample size by the design effect:

Hide

```
# calculate effective sample size
n_samp_eff <- n_samp / Deff
n_samp_eff
```

```
## [1] 9.382689
```

Here we find that the effective sample size is somewhere between 9 and 10, even though the true sample size was 100 per cluster. This provides quite an intuitive way of thinking about within-cluster correlation: even though we have sampled 100 individuals from a cluster, they tend to look very similar to one another so it's really like we've only sampled 9 or 10 people.

Comparing prevalence against a threshold

We now move on to the main motivation of this practical - designing a *pfhrp2/3* deletion study. In 2020, the WHO released a *Master protocol for surveillance of pfhrp2/3 deletions and biobanking to support future research* (https://apps.who.int/iris/handle/10665/331197). This document is an excellent reference tool for anyone intending to conduct a study into *pfhrp2/3* prevalence, and covers everything from laboratory techniques to making an analysis plan. It suggests a clustered survey design with the following features:

- The study is carried out at province level. Multiple clinics (suggested at least 10) are recruited from within the province, making it a **clustered survey design**.
- Suspected malaria cases presenting at clinics are tested by HRP2-based rapid diagnostic test (RDT) and another method, e.g. microscopy. Discordant results are treated as suspected false-negative RDTs, which are then followed up by gene sequencing to establish whether the *pfhrp2/3* gene deletion is present.
- The primary outcome is the prevalence of *pfhrp2/3* deletions in each cluster. These should then be summarised over clusters to produce an estimate of the overall prevalence of deletions at the province level.
- We are specifically interested in whether prevalence is **above or below the 5% threshold**. If above, then the country is advised to consider switching to a non-HRP2-based RDT to reduce the risk of missing clinical episodes.

Here, we will learn one approach for how to analyse this sort of data, and how to design a study taking into account power and logistical considerations.

Statistical testing

Let's start by loading some data from file:

Hide

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

cluster_onegroup
```

```
##    cluster n_samp    p
## 1        1    100 0.23
## 2        2    100 0.07
## 3        3    100 0.00
## 4        4    100 0.05
## 5        5    100 0.29
## 6        6    100 0.06
## 7        7    100 0.03
## 8        8    100 0.19
## 9        9    100 0.00
## 10      10    100 0.06
```

This simulated dataset is intended to represent the results of a *pfhrp2/3* deletion study. For each cluster, we have the number of samples tested and the cluster-level prevalence of deletions.

As in the previous examples, it is useful to think about what would happen if we were to ignore clustering and simply pool our results. We obtain the following 95% CI on the prevalence of deletions:

Hide

```
# pool results over clusters
n_samp_total <- sum(cluster_onegroup$n_samp)
n_samp_pos <- sum(cluster_onegroup$n_samp * cluster_onegroup$p)

# estimate prevalence
p_bar <- n_samp_pos / n_samp_total

# calcualte standard error
SE <- sqrt(p_bar*(1 - p_bar) / n_samp_total)

p_bar + c(-1.96, 1.96)*SE
```

```
## [1] 0.07957225 0.11642775
```

We obtain a fairly precise estimate of the prevalence, with our CI going from around 8% to 12%. What if we were to compare this against the 5% threshold? The appropriate test here is the one-sample z-test, which can by implemented in R using the `prop.test()` function:

Hide

```
# carry out 1-proportion z-test
prop.test(n_samp_pos, n_samp_total, p = 0.05)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  n_samp_pos out of n_samp_total, null probability 0.05
## X-squared = 47.5, df = 1, p-value = 5.5e-12
## alternative hypothesis: true p is not equal to 0.05
## 95 percent confidence interval:
##  0.08062551 0.11853423
## sample estimates:
##     p
## 0.098
```

Q7. Is this result significant? How should the country respond based on these results?

Click For Answer

But, this pooled analysis ignores the clustered nature of the design. Just as in the earlier examples, we need to account for intra-cluster correlation otherwise we risk coming to the wrong conclusions. Using the cluster-level 95% CI, we obtain the following:

Hide

```
# estimate prevalence
n_clusters <- nrow(cluster_onegroup)
p_bar <- mean(cluster_onegroup$p)

# calcualte standard error
SE <- sqrt(var(cluster_onegroup$p) / (n_clusters - 1))

p_bar + c(-1.96, 1.96)*SE
```

```
## [1] 0.031744 0.164256
```

Q8. Does the 95% CI span 5%? What does this suggest about the possibility of prevalence being significantly above this value?

Click For Answer

The appropriate statistical test when comparing against a threshold is the one-sample t-test. This can be implented in R as follows:

Hide

```
# one-sample t-test against 5% threshold
t.test(x = cluster_onegroup$p, mu = 0.05)
```

```
##
##  One Sample t-test
##
## data:  cluster_onegroup$p
## t = 1.4968, df = 9, p-value = 0.1687
## alternative hypothesis: true mean is not equal to 0.05
## 95 percent confidence interval:
##  0.02545405 0.17054595
## sample estimates:
## mean of x
##     0.098
```

Q9. Is this result significant? How should the country respond based on these results?

Click For Answer

Here we have an example where our overall conclusions, and the action that a country takes in response to *pfhrp2/3* deletions, is different depending on our method of analysis. If we incorrectly pool results without accounting for intra-cluster correlation then we can end up making a recommendation about country-wide RDT usage that is not well supported. This highlights the importance of understanding clustering in analysis of survey data.

Power analysis and sample size calculation

How can we power a *pfhrp2/3* deletion study, given what we know about clustering? We start by looking at the theoretical value of our test statistic for the one-sample t-test:

$$t = \frac{p-\mu}{\sqrt{\frac{\sigma^2}{c}}}$$

where p is the true prevalence, σ_c^2 is the variance between clusters and c is the number of clusters. But, remember that the design effect is defined as the variance between clusters divided by the variance we would expect under SRS. So, rearranging this, we can express the variance between clusters as follows:

$$\sigma_c^2 = D_{\text{eff}} \frac{p(1-p)}{n}$$

If we had an assumed prevalence p, a sample size n per cluster, and an assumed design effect D_{eff} then we could work out the appropriate cluster-level variance to use in our t-statistic.

Q10. If the true prevalence of deletions is 10%, and assuming 100 samples per cluster and a design effect of 1.5, what would you expect in terms of the variance between clusters?

Click For Answer

Sometimes it is easier to work in terms of the intra-cluster correlation coefficient (ICC) rather than the design effect. The ICC is often given the mathematical symbol ρ, and is related to the design effect through the formula:

$$D_{\text{eff}} = 1 + (n-1)\rho$$

This leads to the following alternative formula for the variance between clusters:

$$\sigma_c^2 = (1 + (n-1)\rho)\frac{p(1-p)}{n}$$

Q11. If the true prevalence of deletions was 10%, and assuming 100 samples per cluster and an ICC of 0.2, what would you expect in terms of the variance between clusters?

Click For Answer

Finally, we can substitute this value back into our formula for the t-test statistic:

$$t = \frac{p-\mu}{\sqrt{(1+(n-1)\rho)\frac{p(1-p)}{nc}}}$$

We can use this formula to calculate the distribution of the test statistic under the alternative hypothesis, and therefore to calculate power. The function `get_pow_ttest_thresh()` does this power calculation for you:

Hide

```
# calculate power given certain inputs
get_pow_ttest_thresh(p = 0.1, n_samp = 100, n_clusters = 10, ICC = 0.1, p_thresh =
        0.05)
```

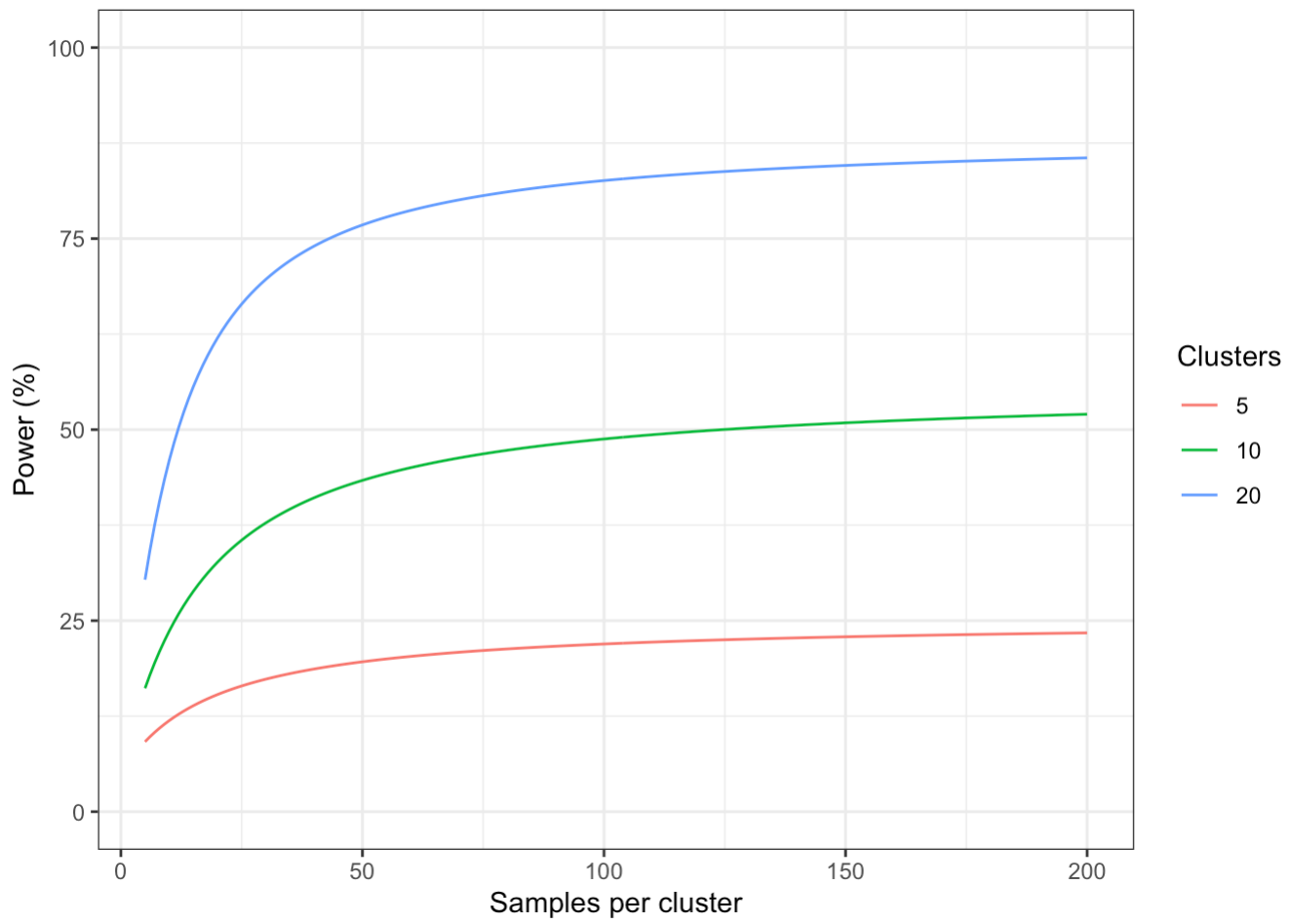
```
## [1] 0.2978355
```

We can use this function to produce power curves for a series of sample sizes and numbers of clusters:

Hide

```
# input parameters
p <- 0.1
ICC <- 0.05
n_samp <- 5:200
n_clusters <- c(5, 10, 20)

# plot power curves
expand_grid(n_clusters, n_samp) %>%
  mutate(pow = get_pow_ttest_thresh(p = p, n_samp = n_samp, n_clusters = n_cluster
        s, ICC = ICC)) %>%
  ggplot() + theme_bw() +
  geom_line(aes(x = n_samp, y = 100*pow, color = as.factor(n_clusters), group = n_
        clusters)) +
  ylim(c(0, 100)) + xlab("Samples per cluster") + ylab("Power (%)") +
  scale_color_discrete(name = "Clusters")
```



Unsurprisingly, we find that power increases as we obtain more samples, and as we recruit more clusters. However, there are also some very interesting things to notice about these power curves. First, unlike the power curves we saw in the previous practical, it does not look like these curves are approaching 100%. We can check this using the following function, which calculates the maximum theoretical power that can be achieved:

Hide

```
# calculate maximum possible power
get_max_pow_ttest_thresh(p = p, n_clusters = n_clusters, ICC = ICC)
```

```
## [1] 0.2511925 0.5568322 0.8850831
```

For the above parameters and for 5 clusters, the maximum power that we could ever achieve would be around 25%. For 10 clusters we can get up to 56% power, and for 20 clusters up to 89%. This tells us that if our aim is to reach 80% power then there is a minimum number of clusters that we *have* to recruit in order for it to be *possible* that we reach this power.

Q12. What is the minimum number of clusters that we have to recruit in the example above for us to have any chance of achieving 80% power?

Click For Answer

The next interesting thing to note from these power curves relates to the total number of samples over all clusters. The following code calculates the sample size required to achieve a give target power, up to a given maximum value:

Hide

```
# define parameters
p <- 0.1
ICC <- 0.01
n_clusters <- c(5, 10, 20)
target_power <- 0.8

# search through sample sizes until reach target power
n_samp_optimal <- expand_grid(n_clusters, n_samp = 5:1e3) %>%
  mutate(pow = get_pow_ttest_thresh(p = p, n_samp = n_samp, n_clusters = n_cluster
        s, ICC = ICC)) %>%
  group_by(n_clusters) %>%
  summarise(n_samp = n_samp[which(pow > target_power)[1]])

n_samp_optimal
```

```
## # A tibble: 3 × 2
##   n_clusters n_samp
##        <dbl>  <int>
## 1          5     NA
## 2         10     55
## 3         20     19
```

In this case, we find that for 5 clusters we cannot achieve the target power within the search range (and in fact it is theoretically impossible for any sample size). For 10 clusters we need 55 samples per cluster, and for 20 clusters we need 19 samples per cluster. Notice that the total number of samples in the 10-cluster case is `10 * 55 = 550`, and in the 20-cluster case is `20 * 19 = 380` So, somewhat counterintuitively, it is **more statistically efficient** to recruit a larger number of clusters, as it means the total number of samples in the entire study goes down.

Q13. Assume the true prevalence of *pfhrp2/3* deletions is 7%, and the intra-cluster correlation is 0.01. What is the total sample size required if we recruit 20 clinics within our province? What improvement do we gain if we manage to recruit a further 10 clinics?

Click For Answer

As a general rule it is always a good idea to **recruit more clusters, not more individuals per cluster**. Not only does this tend be a more efficient strategy, as we have seen above, but it also makes our sampling more **representative** of the population. If we only managed to recruit 3 clinics then we *may* be able to achieve our desired power, but can 3 clinics really capture an entire province? It is quite possible that what we see in these clinics might not be an accurate representation of the province as a whole. This is one of the reasons the WHO master protocol recommends aiming for at least 10 clinics per district.

One important caveat to the above is that recruiting large numbers of clusters can incur costs. Setting up a cluster, training staff etc. all mean that there may be fixed costs that must consider in our overall budget. So while recruiting more clusters may be more *statistically* efficient, it is not necessarily more efficient in terms of the overall study design.

Q14. Using the parameters from question 12, assume that we are designing a trial within a strict budget. The cost of setting up a cluster is estimated at 500 USD. The cost per sample of processing and sequencing is estimated at 20 USD per sample. What number of clusters provides the most cost-effective way of designing the study, assuming we are aiming for 80% power?

Click For Answer

We have seen how clustered sampling can lead to some extra statistical considerations. If we don't account for these issues then we can get misleading results, which in the worst case can lead to incorrect recommendations at a country level. When we account for these issues, we find that it is often desirable to recruit large numbers of clusters, rather than continually sampling more individuals from the same cluster. However, this needs to be balanced against logistical and budget constraints, which ultimately might restrict the number of clusters that are available. Finally, we should note that the analysis methods presented here are just one simple way of analysing clustered survey data. There are many other techniques, for example multi-level modelling, that can have certain advantages but are beyond the scope of this practical.