11.3: Simulating t-tests

We've already seen some code for simulating a \(t\)-test 1000 times, saving the \(p\)-values, and then calculating the proportion of simulations that are significant (p<0.05). It looked like this:

```
save_ps<-length(1000)
for(i in 1:1000){
  group_A <- rnorm(50,100,7)
  group_B <- rnorm(50,105, 7)
  t_results <- t.test(group_A,group_B,var.equal = TRUE)
  save_ps[i] <- t_results$p.value
}
prop_p<-length(save_ps[save_ps<0.05])/1000
print(prop_p)
```

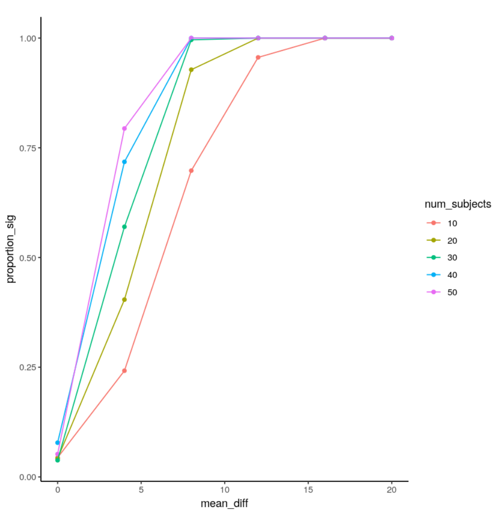
run | restart | restart & run all

```
[1] 0.953
```

You could play around with that, and it would be very useful I think. Is there anything else that we can do that would be more useful? Sure there is. With the above simulation, you have to change N or the mean difference each time to see how proportion of significant experiments turns out. It would be nice to look at a graph where we could vary the number of subjects, and the size of the mean difference. That's what the next simulation does. This kind of simulation can make your computer do some hard work depening on how many simulations you run. To make my computer do less work, we will only run 100 simulations for each parameter. But, what we will do is vary the number of subjects from 10 to 50 (steps of 10), and vary the size of the effect from 0 to 20 in steps of 4.

```
library(ggplot2)
suppressPackageStartupMessages(library(dplyr))
num_sims        <-500
N               <-c(10,20,30,40,50)
mean_difference <-c(0,4,8,12,16,20)
save_ps<-length(num_sims)
all_df<-data.frame()
for(diff in mean_difference){
  for (j in N){
    for(i in 1:num_sims){
      group_A <- rnorm(j,100,7)
      group_B <- rnorm(j,100+diff, 7)
      t_results <- t.test(group_A,group_B,var.equal = TRUE)
      save_ps[i] <- t_results$p.value
    }
    sim_df <-data.frame(save_ps,
                        num_subjects=as.factor(rep(j,num_sims)),
                        mean_diff =rep(diff,num_sims))
    all_df <- rbind(all_df,sim_df)
  }
}
plot_df <- all_df %>%
            dplyr::group_by(num_subjects,mean_diff) %>%
            dplyr::summarise(
              proportion_sig = length(save_ps[save_ps<0.05])/num_sims,
              .groups='drop_last'
            )
ggplot(plot_df, aes(x=mean_diff,
                    y=proportion_sig,
                    group=num_subjects,
                    color=num_subjects))+
  geom_point()+
  geom_line()+
  theme_classic()
```

run restart restart & run all



A graph like this is very helpful to look at. Generally, before we run an experiment, we might not have a very good idea of the size of the effect that our manipulation might cause. Will it be a mean difference of 0 (no effect), or 5, or 10, or 20? If you are doing something new, you just might not have a good idea about this. You would know in general that bigger effects are easier to detect. You would be able to detect smaller and smaller effects if you ran more and more subjects. When you run this kind of simulation, you can vary the possible mean differences and the number of the subjects at the same time, and then see what happens.

When the mean diference is 0, we should get an average of 5%, or (0.05 proportion) experiments being significant. This is what we expect by chance, and it doesn't matter how many subjects we run. When there is no difference, we will reject the null 5% of the time (these would all be type 1 errors).

How about when there is a difference of 4? This a pretty small effect. If we only run 10 subjects in each group, we can see that less than 25% of simulated experiments would show significant results. If we wanted a higher chance of success to measure an effect of this size, then we should go up to 40-50 subjects, that would get us around 75% success rates. If that's not good enough for you (25% failures remember, that's still alot), then re-run the simulation with even more subjects.

Another thing worth pointing out is that if the mean difference is bigger than about 12.5, you can see that all of the designs produce significant outcomes nearly 100% of the time. If you knew this, perhaps you would simply run 10-20 subjects in your experiment, rather than 50. After all, 10-20 is just fine for detecting the effect, and 50 subjects might be a waste of resources (both yours and your participants).