Stat inference notebook

Populations parameters, and sample estimates

Imagine we want to answer the following question : Are men taller than women ? If you have the entire population you could take the height averages for men and women and you would have the answer by comparing averages. But what if we don't have access to the entire population ? We can use a powerful method called statistical inference to have our answer just by looking at some observations rather than the entire population. So the idea is, we take a random sample of men and women and we infer the height of the population for men and women. We will then have sample means for men and women for different samples, and we want to know how close they are to the men and women population means.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(rafalib)

population <- read.csv("data/mice_pheno.csv")
population <- na.omit( population )

#Males
males_chowdiet_population <- filter(population, Sex == "M" & Diet=="chow") %>% select(Bodyweight) %>% u
males_hfdiet_population<- filter(population, Sex == "M" & Diet=="hf") %>% select(Bodyweight) %>% unlist

set.seed(1)
sample_males_chow <- sample(males_chowdiet_population,25)
chow_sample_mean <- mean(sample_males_chow)

set.seed(1)
sample_males_hf <- sample(males_hfdiet_population,25)
hf_sample_mean <- mean(sample_males_hf)

population_mean_difference <- abs(mean(males_chowdiet_population) - mean(males_hfdiet_population))
sample_mean_difference <- abs(chow_sample_mean - hf_sample_mean)

males_popsample_diff <- abs(population_mean_difference-sample_mean_difference)
```

```r
#Females
females_chowdiet_population <- filter(population, Sex == "F" & Diet=="chow") %>% select(Bodyweight) %>%
females_hfdiet_population<- filter(population, Sex == "F" & Diet=="hf") %>% select(Bodyweight) %>% unlis

set.seed(2)
sample_females_chow <- sample(females_chowdiet_population,25)
chow_sample_mean <- mean(sample_females_chow)

set.seed(2)
sample_females_hf <- sample(females_hfdiet_population,25)
hf_sample_mean <- mean(sample_females_hf)

population_mean_difference = abs(mean(females_chowdiet_population) - mean(females_hfdiet_population))
sample_mean_difference = abs(chow_sample_mean - hf_sample_mean)

females_popsample_diff= abs(population_mean_difference-sample_mean_difference)

cat('Difference between population and sample means for Males:',males_popsample_diff)
```

```
## Difference between population and sample means for Males: 1.399884
```

```r
cat('Difference between population and sample means for Females:',females_popsample_diff)
```

```
## Difference between population and sample means for Females: 0.3647172
```

For the females, our sample estimates (means) were smaller, so our estimate for females is closer to the population difference than with males. What is a possible explanation for this ? As we can see below, the population variance of the females is smaller than the population variance of the males; thus, the sample mean has less variability.

```r
stdev_maleschow_population <- popsd(males_chowdiet_population)
stdev_femaleschow_population <- popsd(females_chowdiet_population)

stdev_maleshf_population <- popsd(males_hfdiet_population)
stdev_femaleshf_population <- popsd(females_hfdiet_population)

cat('Population Standard deviation for chow diet: males',stdev_maleschow_population, ', females:',stdev
```

```
## Population Standard deviation for chow diet: males 4.420501 , females: 3.416438
```

```r
cat('Population Standard deviation for high fat diet: males:',stdev_maleshf_population, ', females:',st
```

```
## Population Standard deviation for high fat diet: males: 5.574609 , females: 5.06987
```

The Central Limit theorem The CLT is one of the most frequently used mathematical results in science. It tells us that when the sample size is large, the average $\bar{Y}$ of a random sample follows a normal distribution centered at the population average μ_Y and with standard deviation equal to the population standard deviation σ_Y, divided by the square root of the sample size N. We refer to the standard deviation of the distribution of a random variable as the random variable's standard error.

Thus this theorem tells us that the sample average, which is a random variable, follows a normal distribution: $\bar{X}$~$N(\mu_x, \frac{\sigma_x}{N})$ where μ_x is the population mean, σ_x is the population standard deviation, and N is the sample size. Since the variance is $\frac{\sigma_x}{N}$, we can conclude that when the sample size N gets bigger and bigger the variance will be smaller which will produce a normal distribution with a smaller spread.

If we do an experiment with data coming from a uniform, exponential or even an unknown distribution, by computing the mean for several samples with a minimum size of 30 observations, the means will be normally

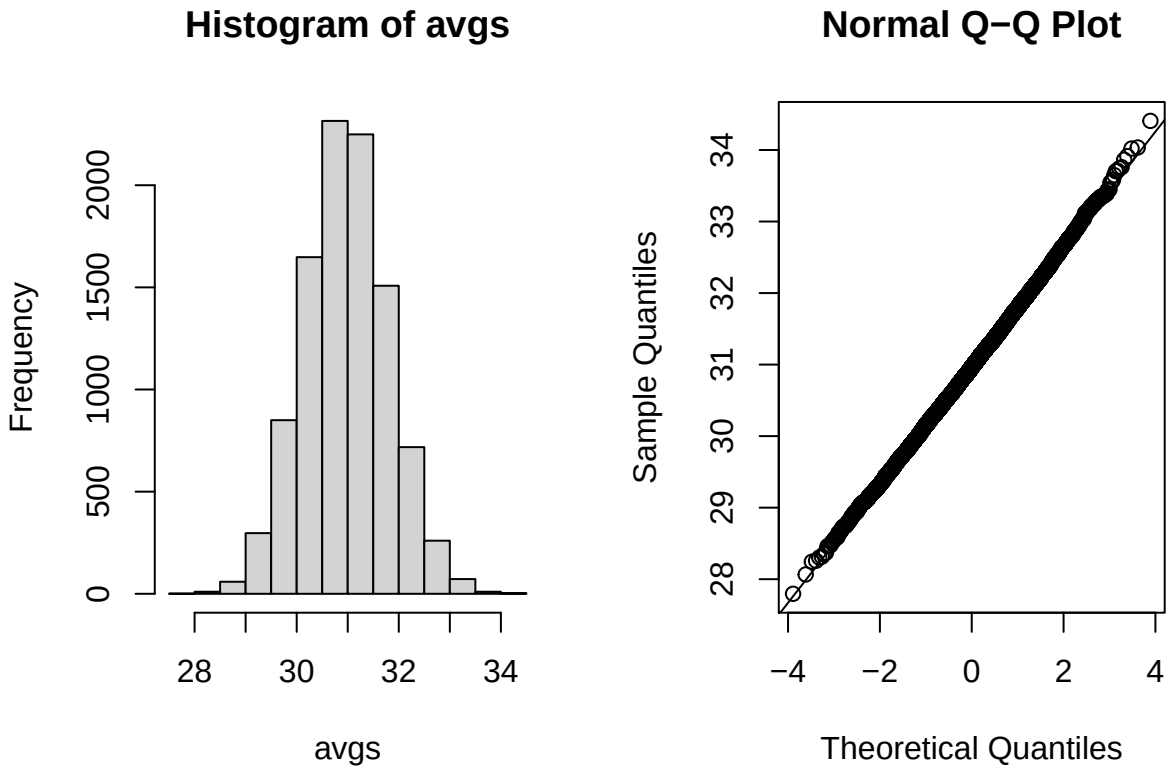
distributed. This is an interesting theorem allowing us to focus on the sample mean normal distribution to make confidence intervals, perform T-test or ANOVA to analyze differences between sample means.

The central limit applies to averages of random variables. Let's explore this concept.

We will now take a sample of size 25 from the population of males on the chow diet. The average of this sample is our random variable. We will use the replicate() function to observe 10,000 realizations of this random variable. Set the seed at 1, then generate these 10,000 averages. Make a histogram and qq-plot of these 10,000 numbers against the normal distribution.

We can see in the qq-plot that, as predicted by the CLT, the distribution of the random variable is very well approximated by the normal distribution.

```
y <- filter(population, Sex=="M" & Diet=="chow") %>% select(Bodyweight) %>% unlist
set.seed(1)
avgs <- replicate(10000, mean( sample(y, 25)))
par(mfrow = c(1,2))
hist(avgs)
qqnorm(avgs)
qqline(avgs)
```




```
cat('Mean:',mean(avgs), ' and standard deviation:',popsd(avgs), 'of averages')
```

```
## Mean: 30.96856  and standard deviation: 0.827082 of averages
```

Once we know the normal distribution is a good approximation, we can start thinking of ways of using it,so that we don't need access to the population data. We can use instead the normal approximation, and compute the mean, the standard deviation and even a p-value without using the entire population.

T tests The goal of a t-test is to compare means of different samples, and see if they are significantly different from each other. We will use two samples X and Y coming from two different diet populations (chow and high fat).

Let's introduce the concept of a null hypothesis. We don't know μ_X nor μ_Y. We want to quantify what the data say about the possibility that the diets have no significant effect: $\mu_X = \mu_Y$. If we use the CLT, then we approximate the distribution of $\bar{X}$ as normal with mean μ_X and standard deviation $\frac{\sigma_X}{\sqrt{M}}$ and the distribution of $\bar{Y}$ as normal with mean μ_Y and standard deviation $\frac{\sigma_Y}{\sqrt{N}}$, with M and N the sample sizes for X and Y respectively, in this case 12. The fact that there is no significant differences between both diets implies that the difference between means of both samples: $\bar{X} - \bar{Y}$ is equal to 0. We described that the standard deviation of this statistic (the standard error) is $SE(\bar{X} - \bar{Y}) = \sqrt{\frac{\sigma_Y}{\sqrt{N}} + \frac{\sigma_X}{\sqrt{M}}}$ and that we estimate the population standard deviations σ_X and σ_Y the population standard deviations, with the sample estimates (the sample standard deviations).

```
library(dplyr)
library(rafalib)
femaleMiceWeights <- read.csv("data/femaleMiceWeights.csv")

X <- filter(femaleMiceWeights, Diet=="chow") %>% select(Bodyweight) %>% unlist
Y <- filter(femaleMiceWeights, Diet=="hf") %>% select(Bodyweight) %>% unlist

xbar <- mean(X)
ybar <- mean(Y)

varx <- var(X)
vary <- var(Y)

size <- length(X)
standard_error <- sqrt((varx/size) + (vary/size))
diff <- (ybar - xbar)

tstat = diff / standard_error
cat("tstat:",tstat)
```

```
## tstat: 2.055174
```

Statistical theory tells us that if we divide a random variable (here the difference between samples means) by its SE, we get a new random variable with an SE of 1. This ratio is what we call the t-statistic. It's the ratio of two random variables and thus a random variable. Once we know the distribution of this random variable, we can then easily compute a p-value.

The CLT tells us that for large sample sizes, both sample averages xbar (mean of chow diet) and ybar (mean of high fat diet) are normal. Statistical theory tells us that the difference of two normally distributed random variables is again normal, so CLT tells us that tstat is approximately normal with mean 0 (the null hypothesis) and SD 1 (we divided by its SE).

So now to calculate a p-value all we need to do is ask: how often does a normally distributed random variable exceed diff? R has a built-in function, pnorm, to answer this specific question. The function pnorm(a) returns the probability that a random variable following the standard normal distribution falls below a. To obtain the probability that it is larger than a, we simply use 1-pnorm(a). We want to know the probability of seeing something as extreme as diff: either smaller (more negative) than -abs(diff) or larger than abs(diff). We call these two regions "tails" and calculate their size:

```
righttail <- 1 - pnorm(abs(tstat))
lefttail <- pnorm(-abs(tstat))
pval <- lefttail + righttail
cat('p value:',pval)
```

```
## p value: 0.0398622
```

In this case, the p-value is smaller than 0.05 and using the conventional cutoff of $\alpha = 0.05$, we would call the

difference between both diets, statistically significant.

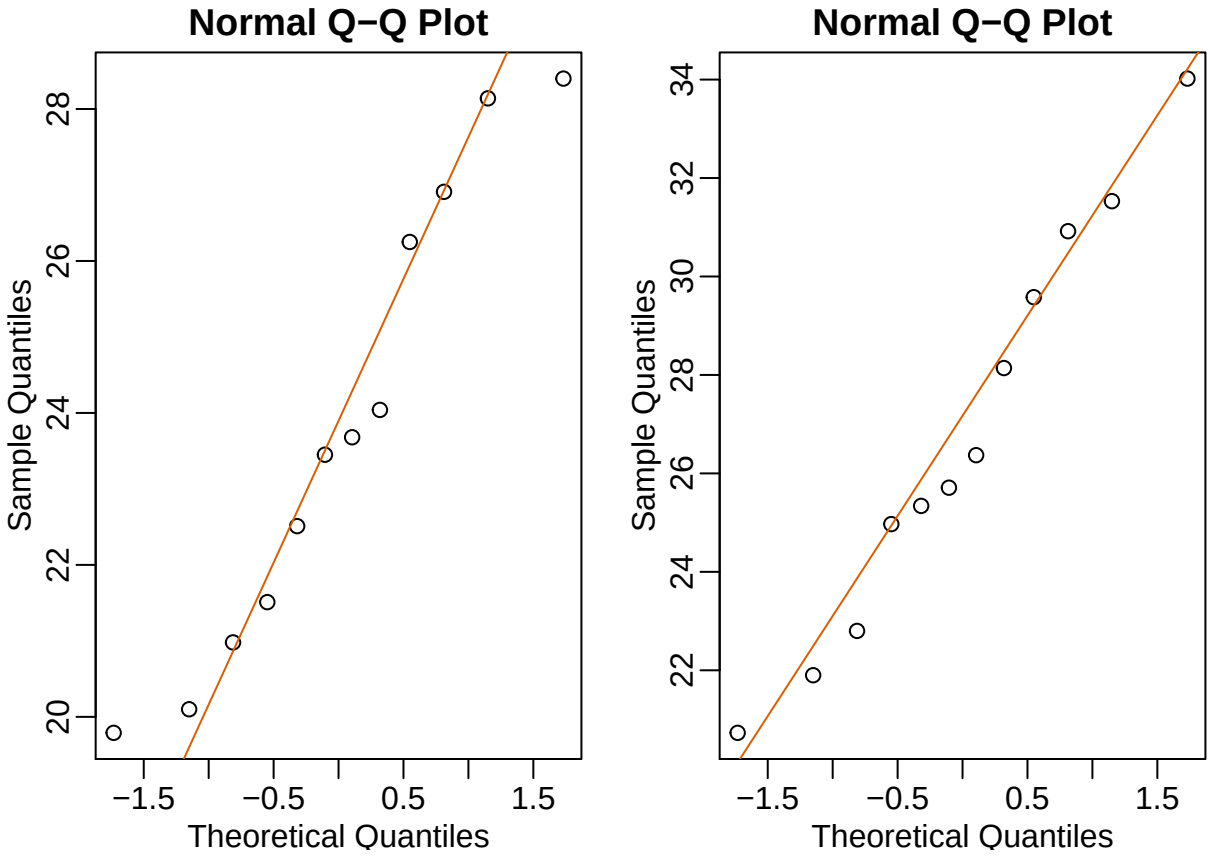
Now there is a problem. CLT works for large samples, but is 12 large enough? A rule of thumb for CLT is that 30 is a large enough sample size (but this is just a rule of thumb). The p-value we computed is only a valid approximation if the assumptions hold, which do not seem to be the case here because our sample sizes are equal to 12 not 30. We need another option.

The t-distribution The t-distribution can be formed by taking many samples (strictly, all possible samples) of the same size from a normal population. For each sample, the same statistic, called the t-statistic, which we will learn more about later, is calculated.The relative frequency distribution of these t-statistics is the t-distribution. It turns out that t-statistics can be computed a number of different ways on samples drawn in a number of different situations and still have the same relative frequency distribution. This makes the t-distribution useful for making many different inferences, so it is one of the most important links between samples and populations used by statisticians

When we have small samples, we can't use the CLT, but we have another option since the data is body weights, we suspect that the population distribution is likely well approximated by normal distribution and that we can use this approximation. If the distribution of the population is normal, then we can work out the exact distribution of the t-statistic without the need for the CLT. And this will be useful for us, because we have small samples of 12 elements, and with small samples, it is hard to check if the population is normal.

Furthermore, we can look at a qq-plot for the samples. This shows that the approximation is at least close:

```
library(rafalib)
mypar(1,2)
qqnorm(X)
qqline(X,col=2)

qqnorm(Y)
qqline(Y,col=2)
```



If we use this approximation, then statistical theory tells us that the distribution of the random variable tstat follows a t-distribution. This is a much more complicated distribution than the normal. The t-distribution has a location parameter like the normal and another parameter called : degrees of freedom. R has a nice function that actually computes everything for us.

```
result <- t.test(X,Y)
cat('T-test pvalue=',result$p.value)
```

```
## T-test pvalue= 0.05299888
```

The p-value is slightly bigger now. This is to be expected because our CLT approximation considered the denominator of tstat practically fixed (with large samples it practically is), while the t-distribution approximation takes into account that the denominator (the standard error of the difference) is a random variable. The smaller the sample size, the more the denominator varies.

It may be confusing that one approximation gave us one p-value and another gave us another, because we expect there to be just one answer. However, this is not uncommon in data analysis. We used different assumptions, different approximations, and therefore we obtained different results.

By reporting only p-values, many scientific publications provide an incomplete story of their findings. With very large sample sizes, scientifically insignificant differences between two groups can lead to small p-values. Confidence intervals are more informative as they include the estimate itself.

Confidence intervals A 95% confidence interval (we can use percentages other than 95%) is a random interval with a 95% probability of having the parameter we are estimating. This parameter could be for example the population mean μ_X. Saying 95% of random intervals will fall on the true value (our definition above) is not the same as saying there is a 95% chance that the true value falls in our interval. To construct confidence intervals, we use the CLT which tells us that: $\sqrt{N}(\bar{X} - \mu_X)/s_X$ follows a normal distribution

with mean 0 and SD 1. This implies that the probability of this event: $-2 \leq \sqrt{N}(\bar{X} - \mu_X)/s_X \leq 2$ is about 95%. Now do some basic algebra to clear out everything and leave μ_X alone in the middle and you get that the following event: $\bar{X} - 2s_X/\sqrt{N} \leq \mu_X \leq \bar{X} + 2s_X/\sqrt{N}$

Here, the edges of the interval $\bar{X} \pm 2s_X/\sqrt{N}$ are random not μ_X. Again, the definition of the confidence interval is that 95% of random intervals will contain the true, fixed value μ_X. For a specific interval that has been calculated, the probability is either 0 or 1 that it contains the fixed population mean μ_X.

Here is an example below:

The population average μ_{chow} is our parameter of interest here, we are interested in estimating this parameter. In practice, we do not get to see the entire population so, as we did for p-values, let's see how we can use samples to do this. Let's start with a sample of size 30:

```
dat <- read.csv("data/mice_pheno.csv")
chowPopulation <- dat[dat$Sex=="F" & dat$Diet=="chow",3]

mu_chow <- mean(chowPopulation)
cat("Population mean:",mu_chow)
```

```
## Population mean: 23.89338
```

```
N <- 30
chow <- sample(chowPopulation,N)
cat('Sample mean:',mean(chow))
```

```
## Sample mean: 24.332
```

We know this is a random variable, so the sample average will not be a perfect estimate. In fact, because in this illustrative example we know the value of the parameter, we can see that they are not exactly the same. A confidence interval is a statistical way of reporting our finding, the sample average, in a way that explicitly summarizes the variability of our random variable.

With a sample size of 30, we will use the CLT. The CLT tells us that $\bar{X}$ (or mean(chow)) follows a normal distribution with mean μ_{chow} (or mean(chowPopulation)) and standard error approximately $\frac{S_{chow}}{\sqrt{N}}$ with S_{chow} the standard deviation of the chow population.

```
se <- sd(chow)/sqrt(N)
cat('Standard error:',se)
```

```
## Standard error: 0.6870825
```

```
Q <- qnorm(1- 0.05/2)
interval <- c(mean(chow)-Q*se, mean(chow)+Q*se )
cat('Confidence interval:',interval)
```

```
## Confidence interval: 22.98534 25.67866
```

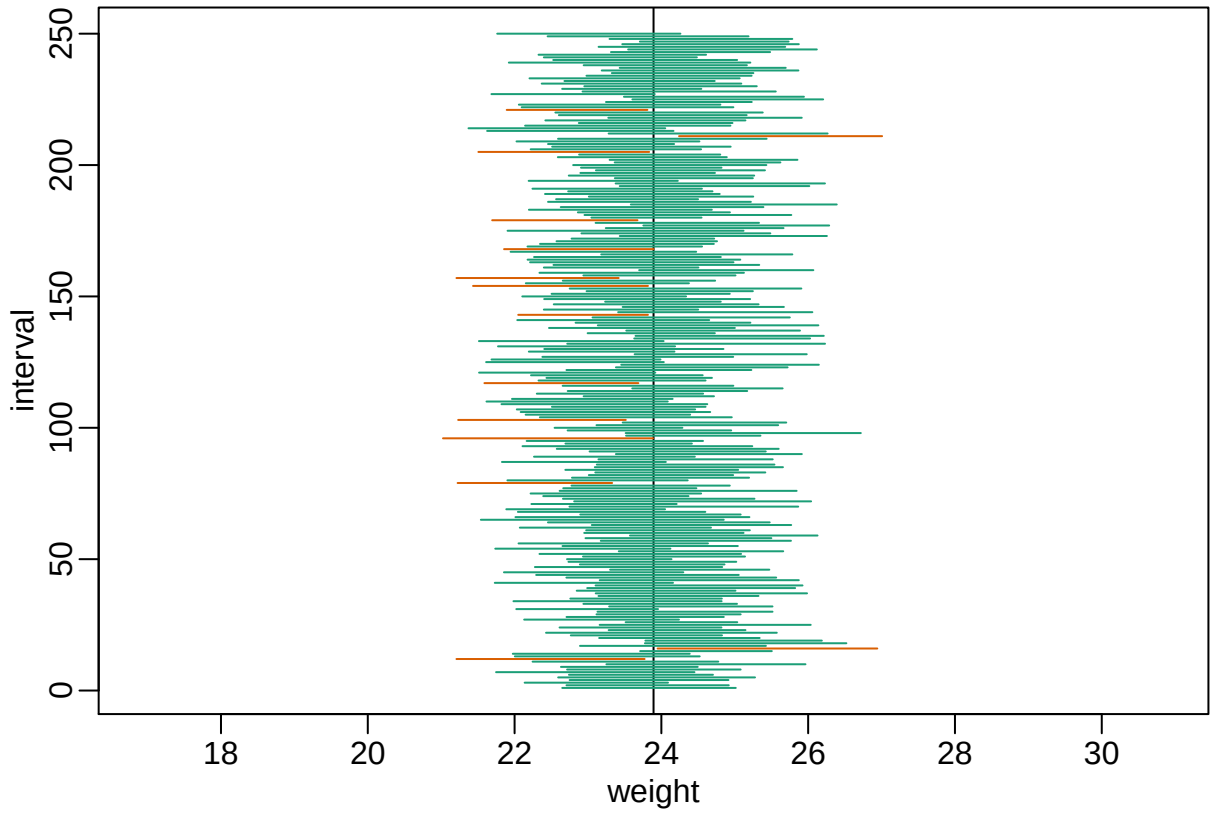
```
isin = interval[1] < mu_chow & interval[2] > mu_chow
cat('Is the population mean (equal to',mu_chow,') in the confidence interval [',interval,']? Answer:',is
```

```
## Is the population mean (equal to 23.89338 ) in the confidence interval [ 22.98534 25.67866 ]? Answer
```

which happens to cover μ_X or mean(chowPopulation). However, we can take another sample and we might not be as lucky. In fact, the theory tells us that we will cover μ_X 95% of the time. Because we have access to the population data, we can confirm this by taking several new samples:

```
library(rafalib)
B <- 250
mypar()
plot(mean(chowPopulation)+c(-7,7),c(1,1),type="n",
```

```
     xlab="weight",ylab="interval",ylim=c(1,B))
abline(v=mean(chowPopulation))
for (i in 1:B) {
  chow <- sample(chowPopulation,N)
  se <- sd(chow)/sqrt(N)
  interval <- c(mean(chow)-Q*se, mean(chow)+Q*se)
  covered <- mean(chowPopulation) <= interval[2] & mean(chowPopulation) >= interval[1]
  color <- ifelse(covered,1,2)
  lines(interval, c(i,i),col=color)
}
```



You can run this repeatedly to see what happens. You will see that in about 5% of the cases, we fail to cover μ_X.

```
babies <- read.table("data/babies.txt", header=TRUE)


bwt.nonsmoke <- filter(babies, smoke == 0) %>% select(bwt) %>% unlist
bwt.smoke <- filter(babies, smoke == 1) %>% select(bwt) %>% unlist

cat('Difference in means:',mean(bwt.nonsmoke)-mean(bwt.smoke))
```

Power calculation

```
## Difference in means: 8.937666
```

We can explore the trade off of power and Type I error concretely using the babies data. Since we have the full population, we know what the true effect size is (about 8.93) and we can compute the power of the test

for true difference between populations.

```
set.seed(1)
N <- 5
dat.ns <- sample(bwt.nonsmoke,N)
dat.s <- sample(bwt.smoke,N)
cat('Is the null hypothesis rejected ? Answer:',t.test(dat.s,dat.ns)$p.value < 0.05)
```

```
## Is the null hypothesis rejected ? Answer: FALSE
```

The p-value is larger than 0.05 so using the typical cut-off, we would not reject. This is a type 2 error where we should reject the null hypothesis but because of the small sample size N we can't find enough evidence to reject the null hypothesis. To reduce type 2 errors, we can use a higher sample size value, or a higher cut-off threshold α. Finding a population for which the null is not true is not a solution here.

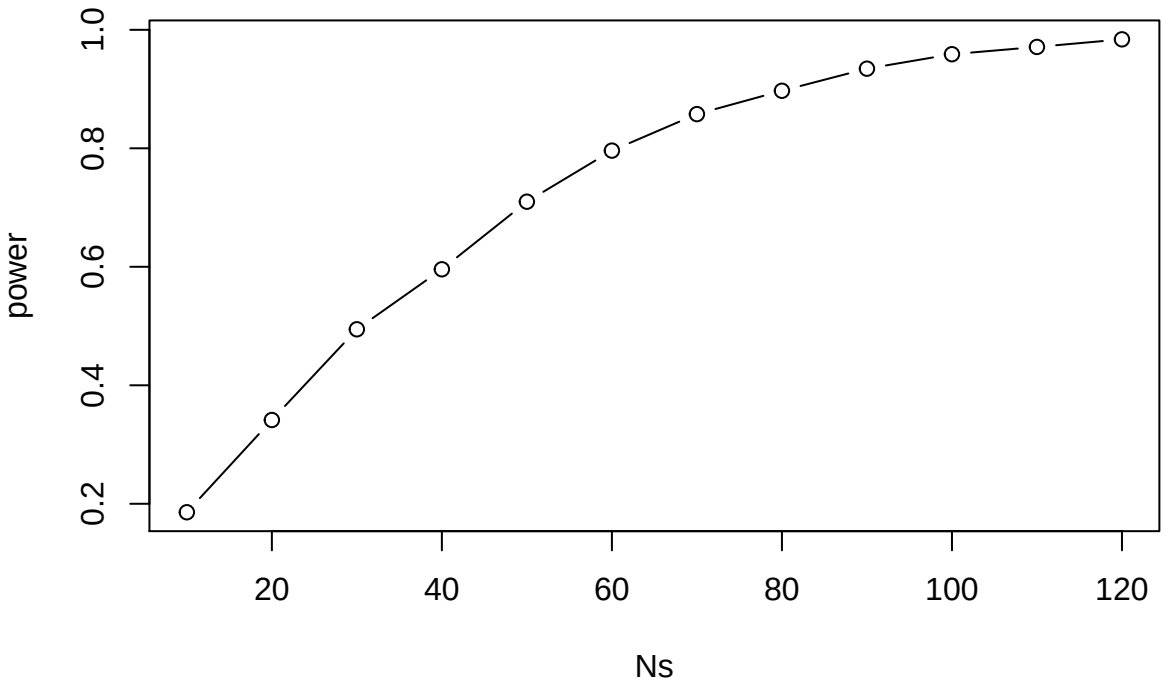
Power is the probability of rejecting the null hypothesis while the alternative hypothesis is true. Thus, we can compute the power by using the replicate() function to repeat population sampling and pvalue computation. After that we can find the proportion of times we rejected the null hypothesis, the power, by applying the mean function.

```
set.seed(1)
N <- 5
dat.ns <- sample(bwt.nonsmoke,N)
dat.s <- sample(bwt.smoke,N)

set.seed(1)
B<-10000
reject <- function(N, alpha=0.05){
  dat.ns <- sample(bwt.nonsmoke,N)
  dat.s <- sample(bwt.smoke,N)
  t.test(dat.s, dat.ns)$p.value < 0.05
}


rejections <- replicate(B,reject(N))
power <- mean(rejections)
cat('Power:',power)
```

```
## Power: 0.096
```

Here the power is very low, but what happens if we use higher sample size values N ?

```
Ns <- seq(10,120,10)
power <- sapply(Ns,function(N){
  rejections <- replicate(B, reject(N))
  mean(rejections)
})

plot(Ns,power,type='b')
```



As we can see when the sample size N increases, power also increases.

Association tests

We want to know if an allele is responsible for a disease. So we will do a Chi-square test to test the null hypothesis h_0 saying that the alleles are not responsible for the disease and thus that there is no difference between them.

To reject or accept the null hypothesis, we need the p-value which we can find in the chi-squared table by using: X^2 and df, the degree of freedom.

Firstly we compute the total for each lines and columns in order to compute the expected value E for each cell in the contingency table. Secondly, we compute $X^2 = \sum \frac{(O-E)^2}{E}$ with 'O' the observed value we have, and 'E' the expected value. After that, we choose a degree of freedom equal to the number of lines (the number of alleles) minus 1, times the number of columns (the number of categories, here sick or not) minus 1 : $df = (2-1) \times (2-1) = 1 \times 1 = 1$ Now that we have X^2 and df we can use them to find our p-value. There are several ways to get a p-value but one way is to look at the chi-squared table (available here: https://www.mathsisfun.com/data/chi-square-table.html). We select the line associated with our degree of freedom $df = 1$, follow along until the closest value of our X^2 and then check the corresponding number in the top row to see the approximate probability ("Significance Level") for that value.

The chisq.test() function do all this for us and we end up with a p-value of 0.067. The latter is slightly greater than $\alpha = 0.05$ so there is not enough evidences to say that alleles are responsible for the appearance of the disease. Thus we can't reject the null hypothesis h_0. This test only works for categorical data (data in categories), such as Gender {Men, Women} or color {Red, Yellow, Green, Blue} etc, but not numerical data such as height or weight.

```
d = read.csv("data/assoctest.csv")

#returns a contingency table with counts in each cell
table <- table(d)
print(table)
```

```
##       case
```

```
## allele  0  1
##      0 17 17
##      1 10 28
```

```
print(chisq.test(table))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table
## X-squared = 3.3437, df = 1, p-value = 0.06746
```

We can also do a fisher test to obtain a p-value which will be the sum of the probabilities of all event equally rare, or rarer.

```
print(fisher.test(table))
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  table
## p-value = 0.05194
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.940442 8.493001
## sample estimates:
## odds ratio
##   2.758532
```

Here the p-value is lower than before but still slightly greater than than $\alpha = 0.05$ so as above we fail to reject the null hypothesis