Sample Size

Sample Size Decisions

ROS defines power as the probability that a particular comparison will achieve "statistical significance" given some assumed true effect size.

Power analysis follows this procedure:

1. hypothesize an effect size 2. assume variance in the data and sample size 3. calculate chance p-value is below threshold

For example assume we are in a one-sample t-test framework.

1. hypothesize effect size: 0.5 *(in other words let $\theta = 1$ and $\theta_0 = 0$)*
2. Set variance to 1 and sample size to 25

3. calculate chance p-value is below threshold. We will use the sample mean ($\bar{X}$) as our test statistic. Under the null, the sampling distribution of our test statistic is $N(0, \frac{1}{25})$. So we set a threshold for sampling distribution such that the probability that $\bar{X}$ exceeds that threshold is α (under a one-sided test). So that would be `qnorm(.95, mean = 0, sd = .2)` $= 0.3289707$.

Finally we need to compute the probability that the sample statistic is less than the threshold value, under the alternative framework. In this scenario $N\left(0.5, \frac{1}{25}\right)$. So we need to calculate the probability that the test statistic is less than the threshold, assuming that the alternative is true. So in R this is `1 - pnorm(qnorm(.95, mean = 0, sd = .2), mean = 0.5, sd = .2)` $= 0.8037649$.

For more details see on power see Christian Stratton and Jenny Green's wonderful Shiny app https://christianstratton.shinyapps.io/PowerApp/

A few points about power:

- **requires** assumptions about things we don't know: effect size and variance of data

- feeds into NHST: p-values and disconnect between statistical significance and practical significance

- The question about how many samples, still typically is as many as you can afford.

An alternative is to design studies to achieve a specified standard error.

In fairly simple situations such as the t-test we set up, with precise estimates of variation in the data, we can get analytic estimates of the standard error which enables a solution for minimum sample size for various levels of standard error.

Design Analysis with Simulation More generally we can use simulation to estimate parameters or interest, such as the standard error as a function of sample size.

Let's reconsider the example we used for a power analysis: a one sample t-test. More generally, the goal is to estimate the population mean.

Goal: design a setting where we can explore the impact of the variation of the data and the number of samples collected on the sampling distribution of the sample statistic.

Note: in this scenario there is a simple analytical solution to the standard error, but the general purpose simulation will always work, even in more complicated scenarios.

Simulation part 1: Simulating data, looking at efficiency of methods

```
num_samples <- 10
var_data <- 1
num_sims <- 1000000

tic('replicate')
data_replicate <- replicate(num_sims, rnorm(num_samples, 0, sqrt(var_data))) # mean not important here
toc()
```

```
## replicate: 8.842 sec elapsed
```

```
tic('loop')
data_loop <- matrix(0,num_sims, num_samples)
for (iter in 1:num_sims){
  data_loop[iter,] <- rnorm(num_samples, 0, sqrt(var_data))
}
toc()
```

```
## loop: 5.11 sec elapsed
```

```
tic('rmnorm')
data_mvnorm <- rmnorm(num_sims, rep(0,num_samples), var_data * diag(num_samples))
toc()
```

```
## rmnorm: 1.652 sec elapsed
```

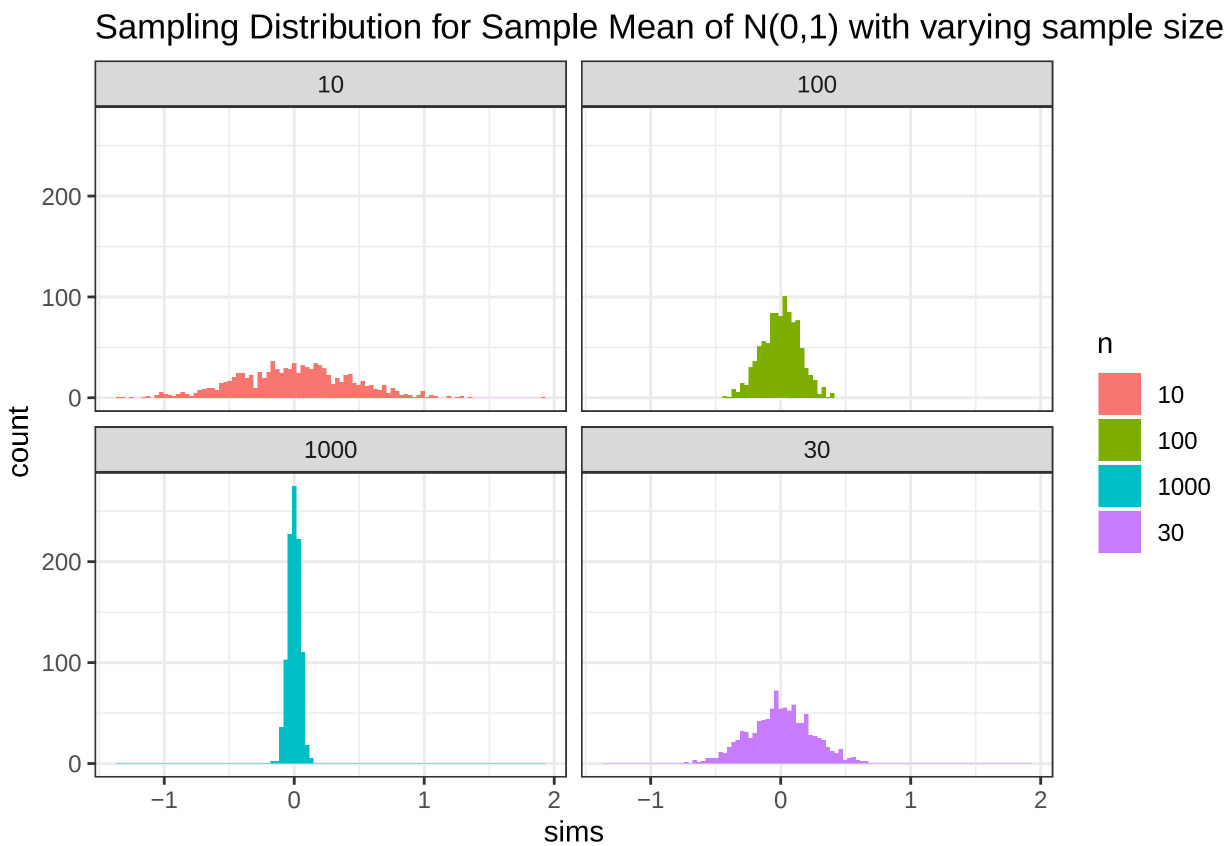
Simulation part 2: Writing a function to return the sampling distribution of the test statistic

```
num_replicates <- 1000
var_data <- 2
num_samples <- 10

simulate_mean_se <- function(num_replicates, var_data, num_samples){
  # function to return standard error for given sample size and data variance
  # inputs:
  #       - num_replicates: number of data sets to simulate
  #       - var_data: variance of the data
  #       - num_samples: number of data points
  # output:
  #       - num_replicates sample means
  return(rowMeans(mnormt::rmnorm(num_replicates, 0, var_data * diag(num_samples))))
}

tibble(sims = c(simulate_mean_se(num_replicates, var_data, 10),
```

```r
                simulate_mean_se(num_replicates, var_data, 30),
                simulate_mean_se(num_replicates, var_data, 100),
                simulate_mean_se(num_replicates, var_data, 1000)),
      n = rep(c('10','30','100','1000'), each = num_replicates)) %>%
        ggplot(aes(x = sims, fill = n)) + geom_histogram(bins = 100) + theme_bw() +
  facet_wrap(~n) +
  ggtitle("Sampling Distribution for Sample Mean of N(0,1) with varying sample size")
```



Simulation part 3: From here we can easily calculate various properties, such as probability of rejecting the hypothesis or standard error of the estimator.

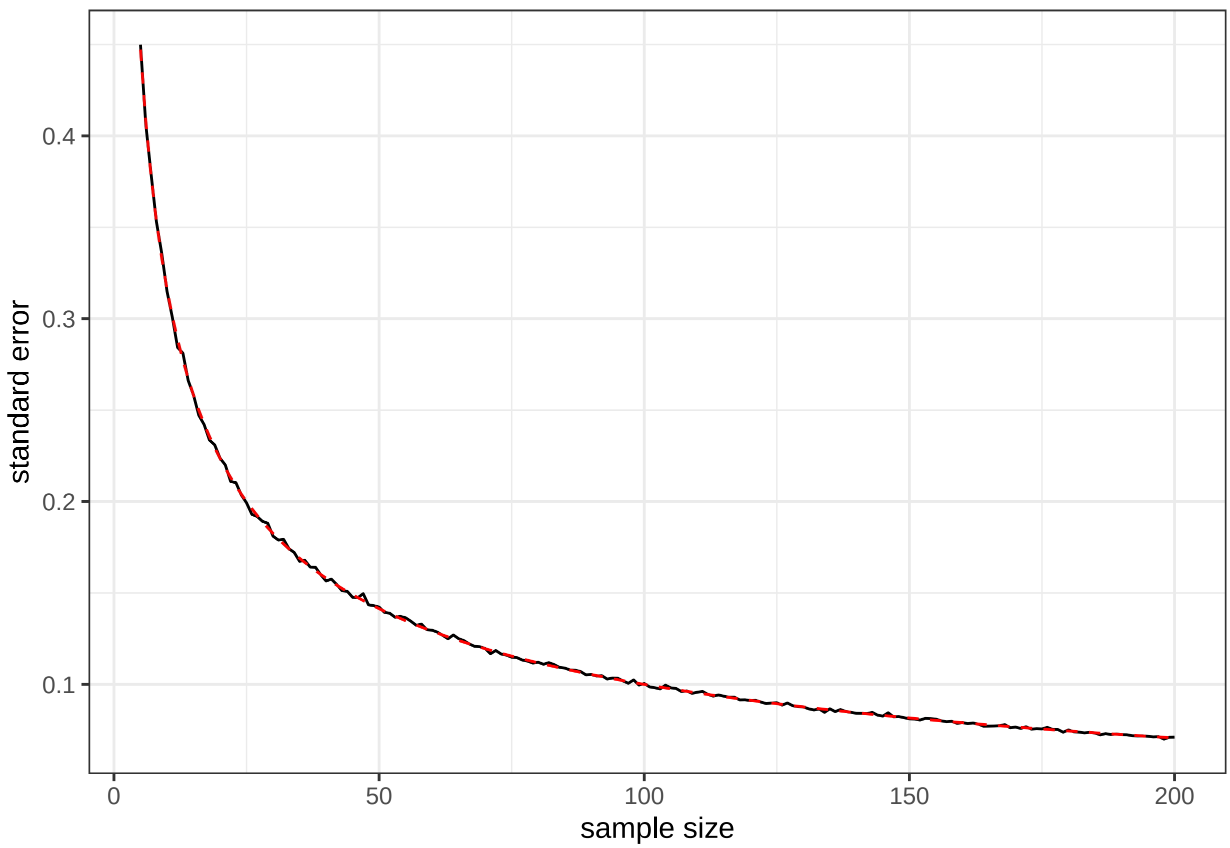
```
num_replicates <- 10000
var_data <- 1
n_seq <- 5:200
numb_n <- length(n_seq)

estimated_se <- rep(0, numb_n)

for (iter in 1:numb_n){
  estimated_se[iter] <- sd(simulate_mean_se(num_replicates, var_data, n_seq[iter]))
}

true_se <- tibble(n_seq = n_seq, se = var_data / sqrt(n_seq))

tibble(n_seq = n_seq, estimated_se = estimated_se) %>%
  ggplot(aes(y=estimated_se, x = n_seq)) +
  geom_line() +
  geom_line(aes(y=se, x = n_seq), inherit.aes = F, data = true_se, color ='red', linetype = 2) +
  theme_bw() + ylab('standard error') + xlab('sample size')
```



Simulation part 4: Often times working with collaborators requires answer many different questions, so a R Shiny application can be a good option.

As an example see: https://andrewhoegh.shinyapps.io/Australian_Samples/