Sample Size

Sample Size Decisions

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

1. hypothesize effect size: 0.5
2. Set variance to 1 and sample size to 25

3. calculate chance p-value is below threshold.

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:

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.

Simulation part 1: ~

```
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: 9.632 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: 4.922 sec elapsed
```

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

```
## rmnorm: 1.598 sec elapsed
```

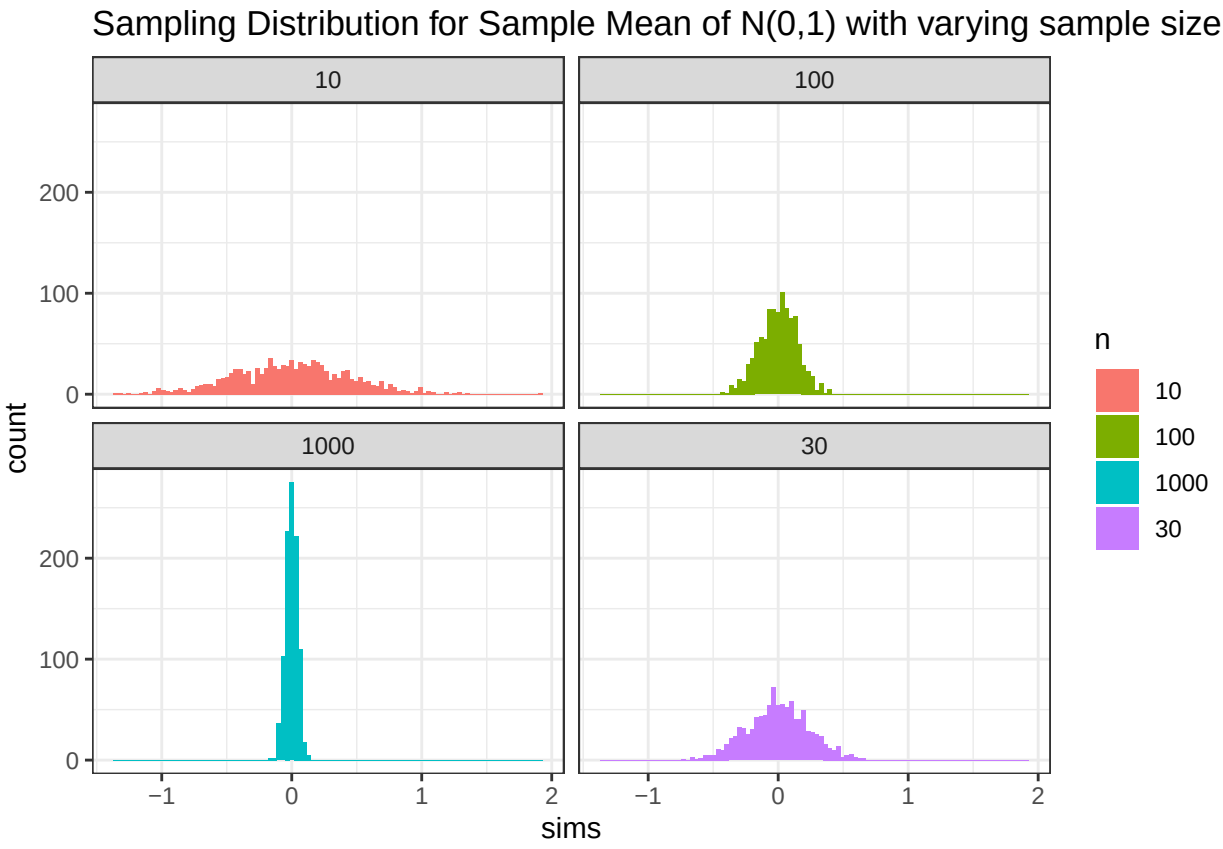
Simulation part 2: ~

```
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),
                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: ~

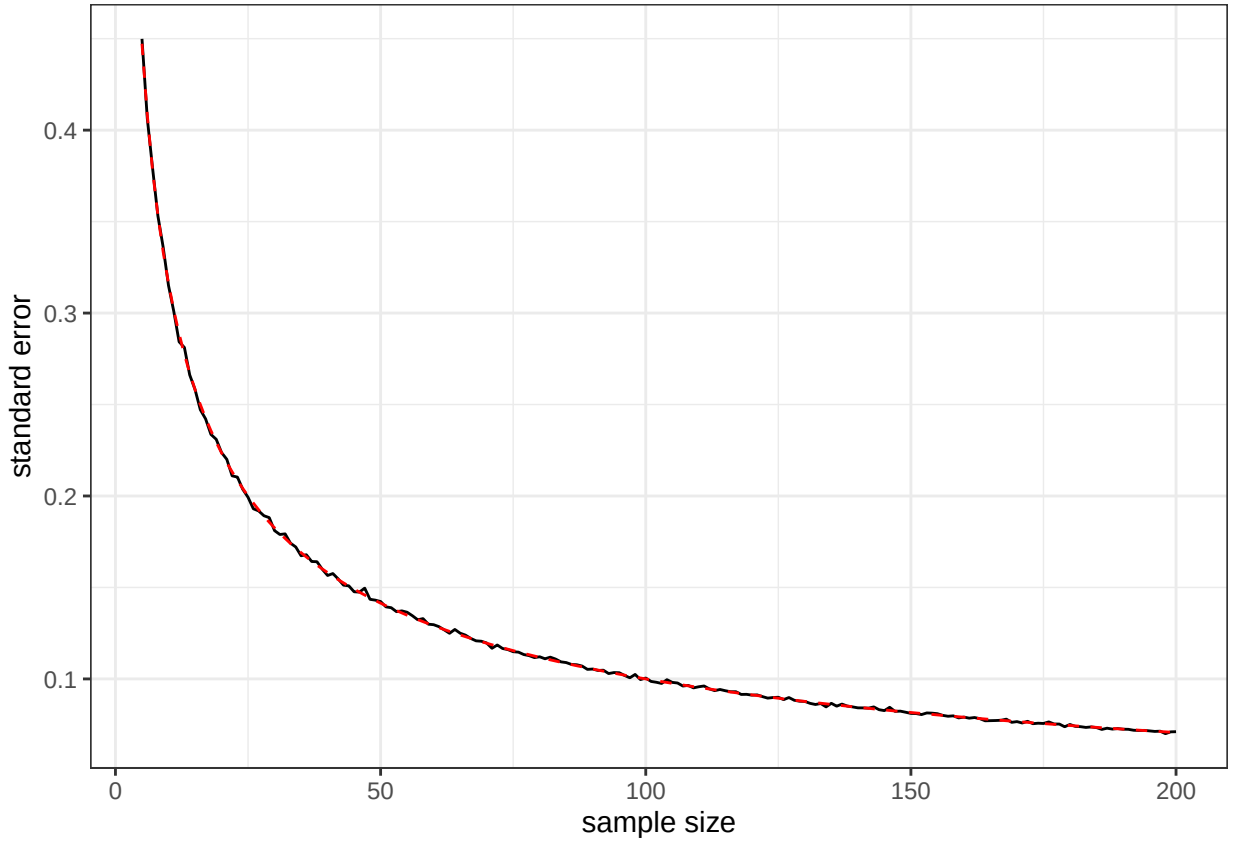
```
num_replicates <- 10000
var_data <- 1
n_seq <- 5:200
numb_n <- length(n_seq)

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/