Central Limit Theorem - Simulation

clear working memory

```
rm(list=ls())
```

loading libraries

```
##
## Attaching package: 'rmutil'

## The following object is masked from 'package:stats':
##
##     nobs

## The following objects are masked from 'package:base':
##
##     as.data.frame, units

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x tidyr::nesting() masks rmutil::nesting()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
# For this exercise, we will use a simulation to see how well the CLT works
# with finite samples in R. Parts (a) and (b) of this question each describe
# a distribution. For each distribution, use 10,000 draws using each of the
# following sample sizes: n = 36, n = 64, n = 100, n = 225, n = 2500,
# and n = 12100. Then discuss how well the normal approximation fits the
# simulated estimates of the means at the critical values of 0.025 and 0.975.

# Suppose that x is binary with Pr(x = 1) = 0.35.


generate_simulated_means <- function(N){
  # Generate the mean and standard deviations of N observations from the specified distribution functi

  # x is binary with Pr(x = 1) = 0.35
```

```r
  # runif(n) returns a vector (length n) of random draw from the uniform distribution U[0,1]
  x <- ifelse(runif(N) < 0.35, 1, 0)

  # put data into data_frame so it is easier to summarize
  data <- tibble(x)

  # get the means for each column
  means <- sapply(data, mean)

  # name the means appropriately
  names(means) <- c("mu1")

  # get the sds for each column
  sds <- sapply(data, sd)

  # name the sds appropriately
  names(sds) <- c("sd1")

  # return the means and standard deviation associated with sample x of size N.
  return(c(means, sds))
  }


# Finding the simulated means and sd for our distributions
# with sample sizes 36, 64, 100, 225, 2500, and 12100
# for each sample size 10,000 replications.

# Using CLT to see how far our observed means were from
# the true means of each distribution.
# Calculating z-scores and then see empirically how many of the means
# were beyond our critical values.

get_zscores <-function(obs_mean, true_mean, obs_sd, N){
  zscores <- (obs_mean - true_mean) / (obs_sd / sqrt(N))
  return( zscores )
}

significance_test <- function(zscores, alpha){
  beyond_critical_point <- as.numeric( zscores > alpha | zscores < -alpha )
  percent_significantly_different <- mean( beyond_critical_point )
  return( percent_significantly_different )
}



monte_carlo <- function(N, reps = 10000){

  replicated_sims <- replicate(reps, generate_simulated_means(N))

  expected_mu <- 0.35
```

```r
  z1 <- get_zscores(replicated_sims['mu1', ], expected_mu, replicated_sims['sd1', ], N)

  sig1 <- significance_test(z1, 0.025)
  print(paste("Percentage of simulated means which were significantly different from"))
  print(paste("sampling distribution at critical point 0.025:", sig1))
  print(paste("      "))

  sig2 <- significance_test(z1, 0.975)
  print(paste("Percentage of simulated means which were significantly different from"))
  print(paste("sampling distribution at critical point 0.975:", sig2))
  print(paste("      "))


}

for (N in c(36, 64, 100, 225, 2500, 12100)){
  print(paste('Starting simulations with samples of size', N))
  monte_carlo(N, 10000)
  print('')
}
```

```
## [1] "Starting simulations with samples of size 36"
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 1"
## [1] "      "
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.975: 0.295"
## [1] "      "
## [1] ""
## [1] "Starting simulations with samples of size 64"
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 1"
## [1] "      "
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.975: 0.295300000000001"
## [1] "      "
## [1] ""
## [1] "Starting simulations with samples of size 100"
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 0.9129"
## [1] "      "
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.975: 0.3426"
## [1] "      "
## [1] ""
## [1] "Starting simulations with samples of size 225"
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 1"
## [1] "      "
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.975: 0.322"
## [1] "      "
## [1] ""
```

```
## [1] "Starting simulations with samples of size 2500"
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 0.9838"
## [1] "      "
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.975: 0.3304"
## [1] "      "
## [1] ""
## [1] "Starting simulations with samples of size 12100"
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 0.977"
## [1] "      "
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.975: 0.3273"
## [1] "      "
## [1] ""
```

```r
# Suppose that x is binary with Pr(x = 1) = 0.97.

generate_simulated_means <- function(N){
   # Generate the mean and standard deviations of N observations from the specified distribution functi

   # x is binary with Pr(x = 1) = 0.97
   # runif(n) returns a vector (length n) of random draw from the uniform distribution U[0,1]
   x <- ifelse(runif(N) < .97, 1, 0)

   # put data into data_frame so it is easier to summarize
   data <- data_frame(x)

   # get the means for each column
   means <- sapply(data, mean)

   # name the means appropriately
   names(means) <- c("mu1")

   # get the sds for each column
   sds <- sapply(data, sd)

   # name the sds appropriately
   names(sds) <- c("sd1")

   # return the means and standard deviation associated with sample x of size N.
   return(c(means, sds))
   }



# Sample sizes 36, 64, 100, 225, 2500, and 12100
# for each sample size 10,000 replications.

# Using CLT to see how far our observed means were from
# the true means of each distribution.
# Calculating z-scores and then see empirically how many of the means
# were beyond our critical values.
```

```r
get_zscores <-function(obs_mean, true_mean, obs_sd, N){
  zscores <- (obs_mean - true_mean) / (obs_sd / sqrt(N))
  return( zscores )
}

significance_test <- function(zscores, alpha){
  beyond_critical_point <- as.numeric( zscores > alpha | zscores < -alpha )
  percent_significantly_different <- mean( beyond_critical_point )
  return( percent_significantly_different )
}



monte_carlo <- function(N, reps = 10000){

  replicated_sims <- replicate(reps, generate_simulated_means(N))

  expected_mu <- 0.97

  z1 <- get_zscores(replicated_sims['mu1', ], expected_mu, replicated_sims['sd1', ], N)

  sig1 <- significance_test(z1, 0.025)
  print(paste("Percentage of simulated means which were significantly different from"))
  print(paste("sampling distribution at critical point 0.025:", sig1))
  print(paste("                    "))

  sig2 <- significance_test(z1, 0.975)
  print(paste("Percentage of simulated means which were significantly different from"))
  print(paste("sampling distribution at critical point 0.975:", sig2))
  print(paste("                    "))


}

for (N in c(36, 64, 100, 225, 2500, 12100)){
  print(paste('Starting simulations with samples of size', N))
  monte_carlo(N, 10000)
  print('')
}
```

```
## [1] "Starting simulations with samples of size 36"
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## i Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 1"
## [1] "                    "
## [1] "Percentage of simulated means which were significantly different from"
```

```
## [1] "sampling distribution at critical point 0.975: 0.428900000000001"
## [1] "                         "
## [1] ""
## [1] "Starting simulations with samples of size 64"
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 1"
## [1] "                         "
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.975: 0.2684"
## [1] "                         "
## [1] ""
## [1] "Starting simulations with samples of size 100"
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 0.7704"
## [1] "                         "
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.975: 0.272099999999999"
## [1] "                         "
## [1] ""
## [1] "Starting simulations with samples of size 225"
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 1"
## [1] "                         "
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.975: 0.3463"
## [1] "                         "
## [1] ""
## [1] "Starting simulations with samples of size 2500"
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 0.9576"
## [1] "                         "
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.975: 0.3549"
## [1] "                         "
## [1] ""
## [1] "Starting simulations with samples of size 12100"
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.025: 0.9789"
## [1] "                         "
## [1] "Percentage of simulated means which were significantly different from"
## [1] "sampling distribution at critical point 0.975: 0.339"
## [1] "                         "
## [1] ""
```

Observations

Central Limit theorem is interpreted here. 1. As we increase N, the percentage of sample means that have a z-score below -0.025 and above 0.025 is ~99%. 2. For critical point z = 0.975: As we increase N, the percentage of sample means that have a z-score below -0.975 and above 0.975 is ~34%, which means 66% of the sample means are between z score of 0.975. These simulation results are in accordance with a typical normal distribution where almost 68% of sample means lie within a z-score of 1 and where many sample means fall outside the z-score of 0.025 as the interval defined by the same is very very small.