Birthday

Cameron Bale

1/19/2019

Load library.

```
#install.packages('tidyverse')
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------
------------ tidyverse 1.2.1 --

## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0

## -- Conflicts ------------------------------------------------------------
------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Create a function, 'birthday', which reports the monte carlo simulatation estimated probability of having a shared birthday in 'n' people. You can specify how many repetitions you want in the simulation using the 'n_reps' argument (default is 1000).

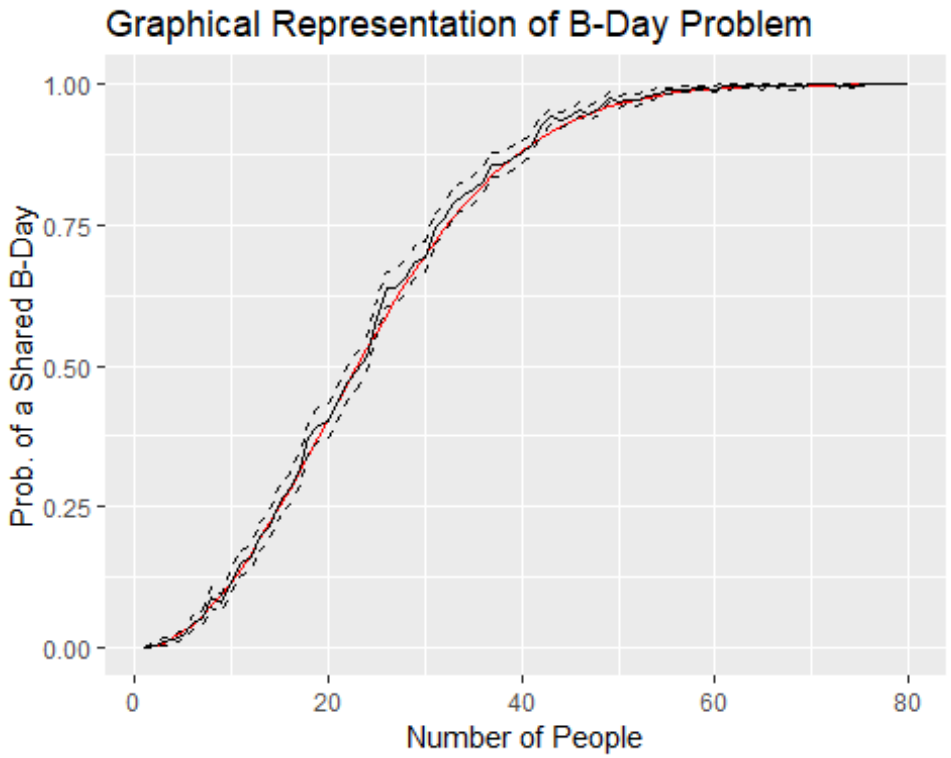
```
birthday <- function (n, n_reps = 1000) {
  # create emply duplicate vector
  num_dups <- 0
  # sample 'n_reps' times, each time noting duplication in 'num_dups'
  for (i in 1:n_reps) {

    b_days <- sample(1:365, n, replace = TRUE)

    if (length(b_days) != length(unique(b_days))) {

      num_dups <- num_dups + 1

    }

  }
  # get probability of shared birthday from all 'n_reps' simulations
  avg_dups <- num_dups / n_reps
  # create confidence interval for the probability
  prob_ci <- avg_dups + c(-1, 1) * qnorm(0.975) * sqrt(avg_dups * (1 -
avg_dups)/n_reps)
```

```
  stats <- append(avg_dups, prob_ci)

  names(stats) <- c('Prob. of Shared B-Day', 'Lower Bound', 'Upper Bound')

  return(stats)

}

birthday(23, 1000)
```

```
## Prob. of Shared B-Day           Lower Bound           Upper Bound
##             0.4990000             0.4680103             0.5299897
```

Plotting monte carlo probability and true probability.

```
n <- 1:80
# using a taylor series approximation of probability of sharing a birthday
true_prob_approx <- sapply(n, function (x) 1 - exp(-x * (x - 1) / 730))
# store monte carlo estimates for 1 to 80 people
mc_estimates <- lapply(n, birthday)
# store means and confidence interval bounds
mc_means <- sapply(n, function (x) mc_estimates[[x]][[1]])
mc_lower <- sapply(n, function (x) mc_estimates[[x]][[2]])
mc_upper <- sapply(n, function (x) mc_estimates[[x]][[3]])
# bring data together into one dataframe
graph_data <- tibble('True' = true_prob_approx,
                     'Mean' = mc_means,
                     'Lower' = mc_lower,
                     'Upper' = mc_upper,
                     'x' = n)
# graph data
graph_data %>%
  ggplot(aes(x = n, y = True)) +
  geom_line(color = 'red') +
  geom_line(aes(x = x, y = Mean)) +
  geom_line(aes(x = x, y = Lower), linetype = 2) +
  geom_line(aes(x = x, y = Upper), linetype = 2) +
  labs(x = 'Number of People',
       y = 'Prob. of a Shared B-Day',
       title = 'Graphical Representation of B-Day Problem')
```


Graphical Representation of B-Day Problem
Prob. of a Shared B-Day
1.00
0.75
0.50
0.25
0.00
0
20
40
60
80
Number of People