All_Four

Cameron Bale

1/14/2019

Function that Simulates Buying Cereal

Create a function that simulates purchasing cereal boxes until you have one of each toy. Because we are working with means, and the sample size is large, by the central limit theorem, we can assume the monte carlo error will be approximately normal.

```
cereal_sim <- function (purchase_options, probabilities, confidence) {
  # create a function which calculates the number of boxes needed to collect
1 of each toy
  box_buyer <- function (purchase_options, probabilities) {
    # create vector for 10,000 purchases
    samp <- sample(purchase_options, size = 10000, replace = TRUE, prob =
probabilities)
    # find the first instance of each toy and report the maximum instance.
This represents
    # how many boxes it took to get 1 of each toy
    draws <- sapply(purchase_options, function (x) min(which(samp ==
purchase_options[x])))
    # return the number of boxes purchased to get 1 of each toy
    return(max(draws))
  }
  # simulate obtaining 1 of each toy 1000 times
  sim <- sapply(1:10000, function (x) box_buyer(purchase_options,
probabilities))
  # calculate average number of boxes purchased to get at least one of each
toy
  avg_boxes <- mean(sim)
  names(avg_boxes) <- 'Average # Boxes'
  # calculate proportion of customers who purchased at least 14 boxes
  prop_14 <- length(sim[sim >= 14]) / length(sim)
  names(prop_14) <- 'Proportion Purchased >= 14'
  # calculate confidence interval for mean
  alpha <- 1 - confidence
  table_value <- qnorm(alpha/2, lower.tail = FALSE)
  sd <- sd(sim)
  mean_ci <- avg_boxes + c(-1, 1) * table_value * sd / sqrt(length(sim))
  names(mean_ci) <- c('Lower Bound', 'Upper Bound')
  avg_boxes <- append(avg_boxes, mean_ci)
  # calculate confidence interval for proportion
  prop_14_ci <- prop_14 + c(-1, 1) * table_value * sqrt(prop_14 * (1 -
```

```r
prop_14) / length(sim))
  names(prop_14_ci) <- c('Lower Bound', 'Upper Bound')
  prop_14 <- append(prop_14, prop_14_ci)
  all_stats <- append(avg_boxes, prop_14)
  return(all_stats)
}
```

Scenario 1:

Toys have Equal Probabilities

```r
set.seed(42)
cereal_sim(c(1, 2, 3, 4), c(1/4, 1/4, 1/4, 1/4), .95)
```

```
##             Average # Boxes                   Lower Bound
##                  8.32740000                    8.25340510
##                 Upper Bound Proportion Purchased >= 14
##                  8.40139490                    0.09530000
##                 Lower Bound                   Upper Bound
##                  0.08954498                    0.10105502
```

On average, a consumer will need to buy 8.39 boxes of cereal to collect at least one of each toy. A 95% confidence interval for the mean is (8.16, 8.62). The proportion of customers who had to purchase at least 14 boxes of cereal to get at least 1 of each toy is 0.10. A confidence interval for this proportion is (0.08, 0.11).

Scenario 2:

Toys have Different Probabilities

```r
set.seed(42)
cereal_sim(c(1, 2, 3, 4), c(.1, .25, .25, .4), .95)
```

```
##             Average # Boxes                   Lower Bound
##                  12.0693000                    11.9053227
##                 Upper Bound Proportion Purchased >= 14
##                  12.2332773                     0.2894000
##                 Lower Bound                   Upper Bound
##                   0.2805119                     0.2982881
```

On average, a consumer will need to buy 12.13 boxes of cereal to collect at least one of each toy. A 95% confidence interval for the mean is (11.59, 12.67). The proportion of customers who will have to purchase at least 14 boxes of cereal to get at least one of each toy is 0.3. A confidence interval for this proportion is (0.26, 0.32).