MT5763_2_220021614

Nico Herrig

2022-10-15

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Problem 1

Description: The problem presents two variables X and Y, where X ~ N(μ = 4, σ^2 = 10) and Y ~ U(a = 2, b = 8).
Compute $Pr(X > Y)$ and use bootstrapping to to derive the sampling distribution for your estimate of $Pr(X > Y)$. Show how the sample variance of this sampling distribution changes as a function of the number of Monte Carlo simulations.

For the underlying problem, a sample containing 10000 random deviates from X ~ N(μ = 4, σ^2 = 10) and Y ~ U(a = 2, b = 8) is used. To simulate "real-world conditions", the solution is obtained from only the below given vectors for X and Y.

```r
set.seed(0911)

X <- rnorm(10000, mean = 4, sd = sqrt(10))
Y <- runif(10000, min = 2, max = 8)

Pr_hat <- sum(X > Y) / 10000

print(Pr_hat)
```

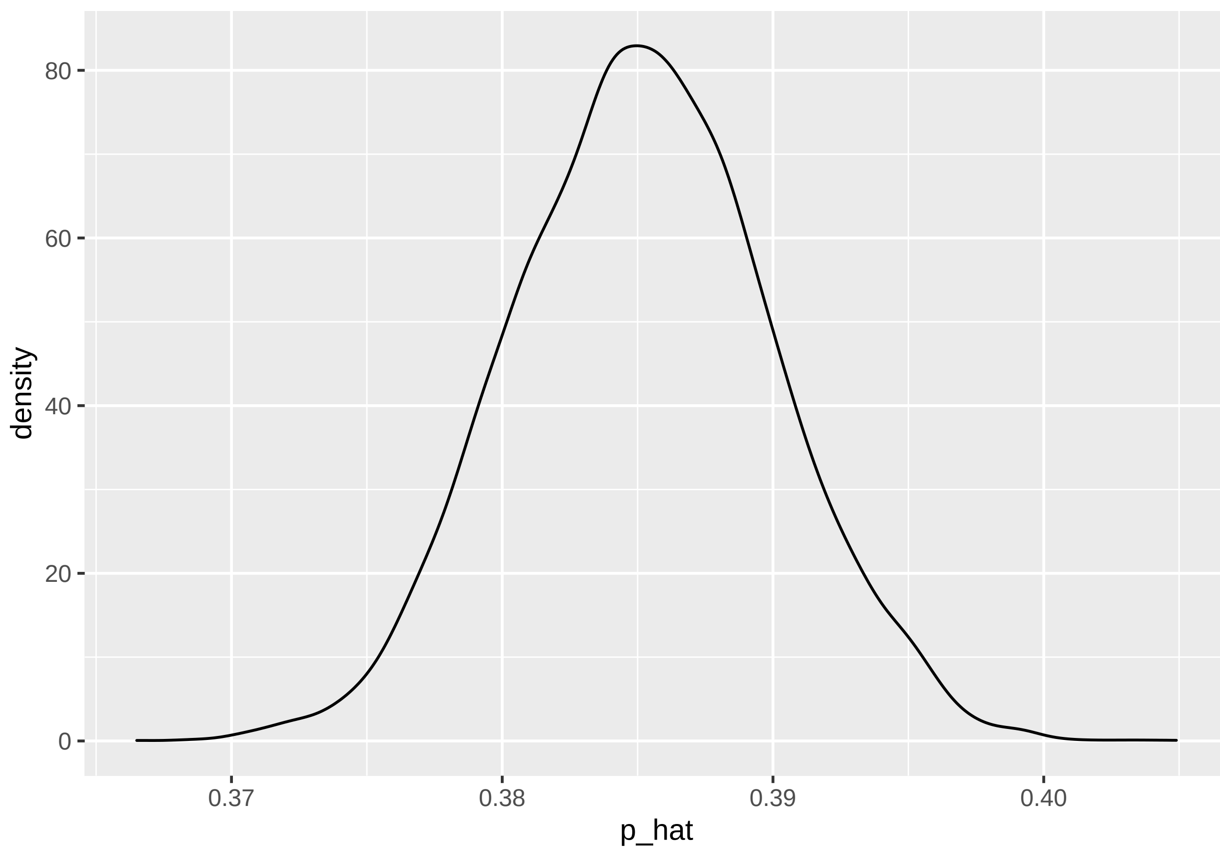
```
## [1] 0.3852
```

Calculating $\bar{P}r(X > Y)$ from the initial sample without any further methods, we derive a value of 0.3852 A non-parametric bootstrapping is now used to simulate the distribution of $\bar{P}r(X > Y)$ from the given sample.

```r
bootstrap <- function(n_bootstraps, vec1 = X, vec2 = Y) {

  vector_prob <- rep(NA, times = n_bootstraps)

  for (i in 1 : n_bootstraps) {
    X_resampled <- vec1[sample(1 : length(vec1), length(vec1), replace = TRUE)]
    Y_resampled <- vec2[sample(1 : length(vec2), length(vec2), replace = TRUE)]

    vector_prob[i] <- sum(X_resampled > Y_resampled) / length(X)
  }
  return(vector_prob)

}
```

The bootstrap algorithm re-samples the vectors X and Y *with replacement* and calculates the resulting $\bar{P}r_i(X > Y)$, and repeating this procedure n times. The funtion puts out a vector with n generated probabilities.
Using the bootstrap function, we now can evaluate the distribution for $Pr(X > Y)$

```r
df_probabilities <- data.frame(p_hat =bootstrap(n_bootstraps = 10000))

df_probabilities %>%
ggplot(mapping = aes(p_hat)) +
  geom_density()
```



```r
MC_sim <- function(runs) {

}
```