ANSWER for EX2.

Jiaqi Wang

2025-09-18

Q1

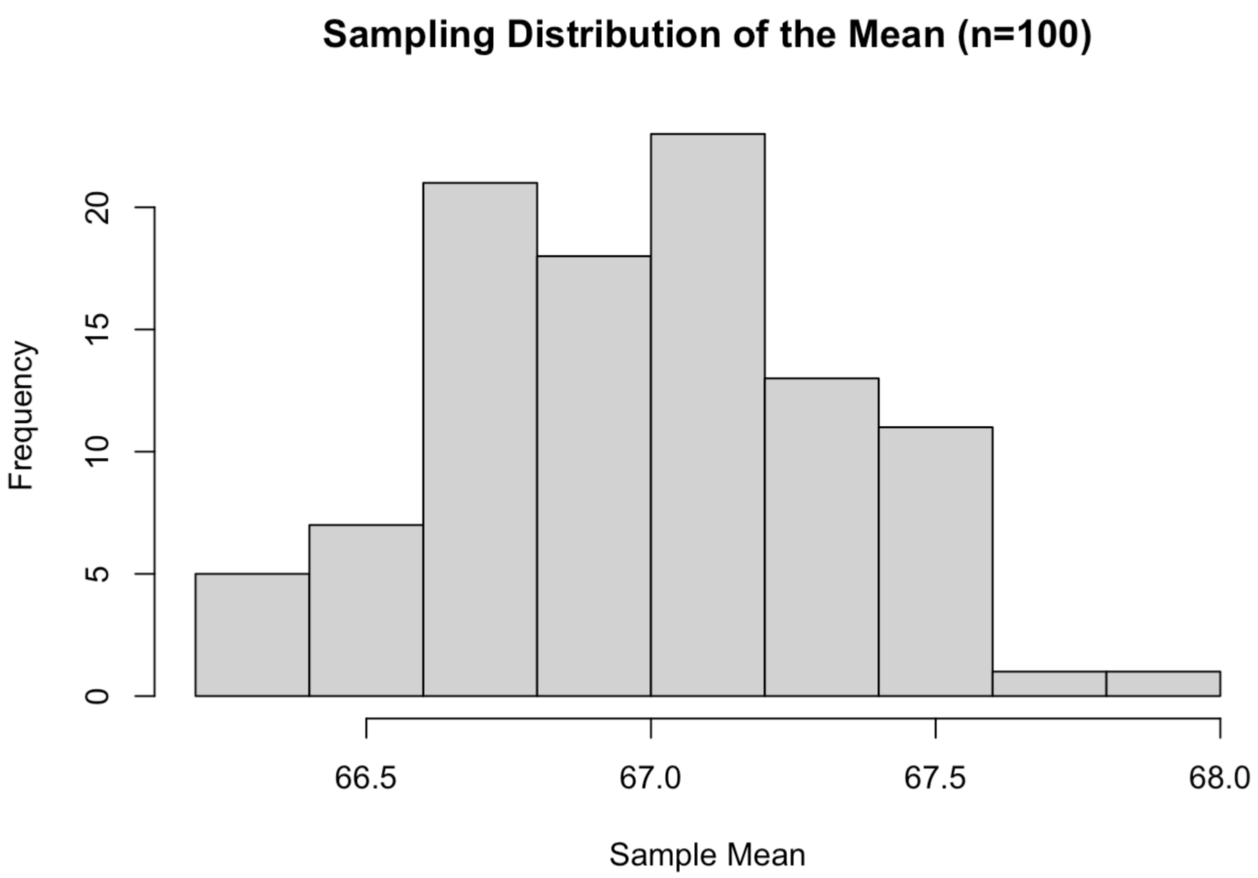
```
#Parameters
mu = 67
sd = 3.5
n = 100

set.seed(10935)
#Simulate B samples, each of size n, from N(mu, sd^2)
height_samples = sapply(1:100, function(x)
  rnorm(n, mu, sd))
```

Q2

```
#Sample means across the B samples
sample_mean <- colMeans(height_samples)

hist(sample_mean, main = "Sampling Distribution of the Mean (n=100)",
     xlab = "Sample Mean")
```



```
mean(sample_mean)
```

```
## [1] 66.98327
```

```
sd(sample_mean)
```

```
## [1] 0.3366694
```

```
sample_sd <- apply(height_samples,
                   2, sd)
sample_sd
```

```
##   [1] 3.427964 3.849204 3.332745 3.497176 3.298559 3.305679 3.350292 3.148000
##   [9] 3.623670 3.441794 3.514039 3.440013 3.255937 3.157225 3.760637 3.759610
##  [17] 3.317468 3.745110 3.454654 3.265977 3.628548 3.213060 3.216942 3.856517
##  [25] 3.534152 3.336856 3.256161 3.224509 3.889919 3.366699 3.113331 3.576201
##  [33] 3.670165 3.368174 3.854601 3.100444 3.289804 3.679158 3.434859 3.777309
##  [41] 3.222927 3.701612 3.565713 3.975732 3.429897 3.559582 3.027391 3.543220
##  [49] 3.601681 3.134940 3.642719 3.525634 4.013635 3.453504 4.087998 3.290100
##  [57] 3.158990 3.915627 3.396414 3.436121 3.287392 3.626695 3.480595 3.342981
##  [65] 3.317116 3.432059 3.615917 3.719834 3.361261 3.184052 3.138380 3.519019
##  [73] 3.599622 3.513977 3.190842 3.418533 3.808131 3.431014 3.367724 3.579981
##  [81] 3.539915 3.563919 3.559479 3.268613 3.016346 3.241813 3.445651 3.306685
##  [89] 3.105444 3.890683 3.529456 3.486970 3.059229 3.308045 3.600756 3.718865
##  [97] 3.000663 3.345454 3.293473 3.373497
```

2.1 Approximately normal, centered near 67, with a tight spread.

2.2 Empirically ≈ 67. Theoretically expected = 67.

2.3 Empirically ≈ 0.35; theoretical SD = 0.35

Q3

```
sample_se <- apply(height_samples,
                   2, function(x) sd(x)/sqrt(n))
sample_se
```

```
##   [1] 0.3427964 0.3849204 0.3332745 0.3497176 0.3298559 0.3305679 0.3350292
##   [8] 0.3148000 0.3623670 0.3441794 0.3514039 0.3440013 0.3255937 0.3157225
##  [15] 0.3760637 0.3759610 0.3317468 0.3745110 0.3454654 0.3265977 0.3628548
##  [22] 0.3213060 0.3216942 0.3856517 0.3534152 0.3336856 0.3256161 0.3224509
##  [29] 0.3889919 0.3366699 0.3113331 0.3576201 0.3670165 0.3368174 0.3854601
##  [36] 0.3100444 0.3289804 0.3679158 0.3434859 0.3777309 0.3222927 0.3701612
##  [43] 0.3565713 0.3975732 0.3429897 0.3559582 0.3027391 0.3543220 0.3601681
##  [50] 0.3134940 0.3642719 0.3525634 0.4013635 0.3453504 0.4087998 0.3290100
##  [57] 0.3158990 0.3915627 0.3396414 0.3436121 0.3287392 0.3626695 0.3480595
##  [64] 0.3342981 0.3317116 0.3432059 0.3615917 0.3719834 0.3361261 0.3184052
##  [71] 0.3138380 0.3519019 0.3599622 0.3513977 0.3190842 0.3418533 0.3808131
##  [78] 0.3431014 0.3367724 0.3579981 0.3539915 0.3563919 0.3559479 0.3268613
##  [85] 0.3016346 0.3241813 0.3445651 0.3306685 0.3105444 0.3890683 0.3529456
##  [92] 0.3486970 0.3059229 0.3308045 0.3600756 0.3718865 0.3000663 0.3345454
##  [99] 0.3293473 0.3373497
```

```
t_crit   <- qt(0.975, df = n - 1)
lower_ci <- sample_mean - t_crit*sample_se
upper_ci <- sample_mean + t_crit*sample_se
covered <- (lower_ci <= mu) & (upper_ci >= mu)
coverage_percent <- mean(covered) * 100
coverage_percent
```

```
## [1] 97
```

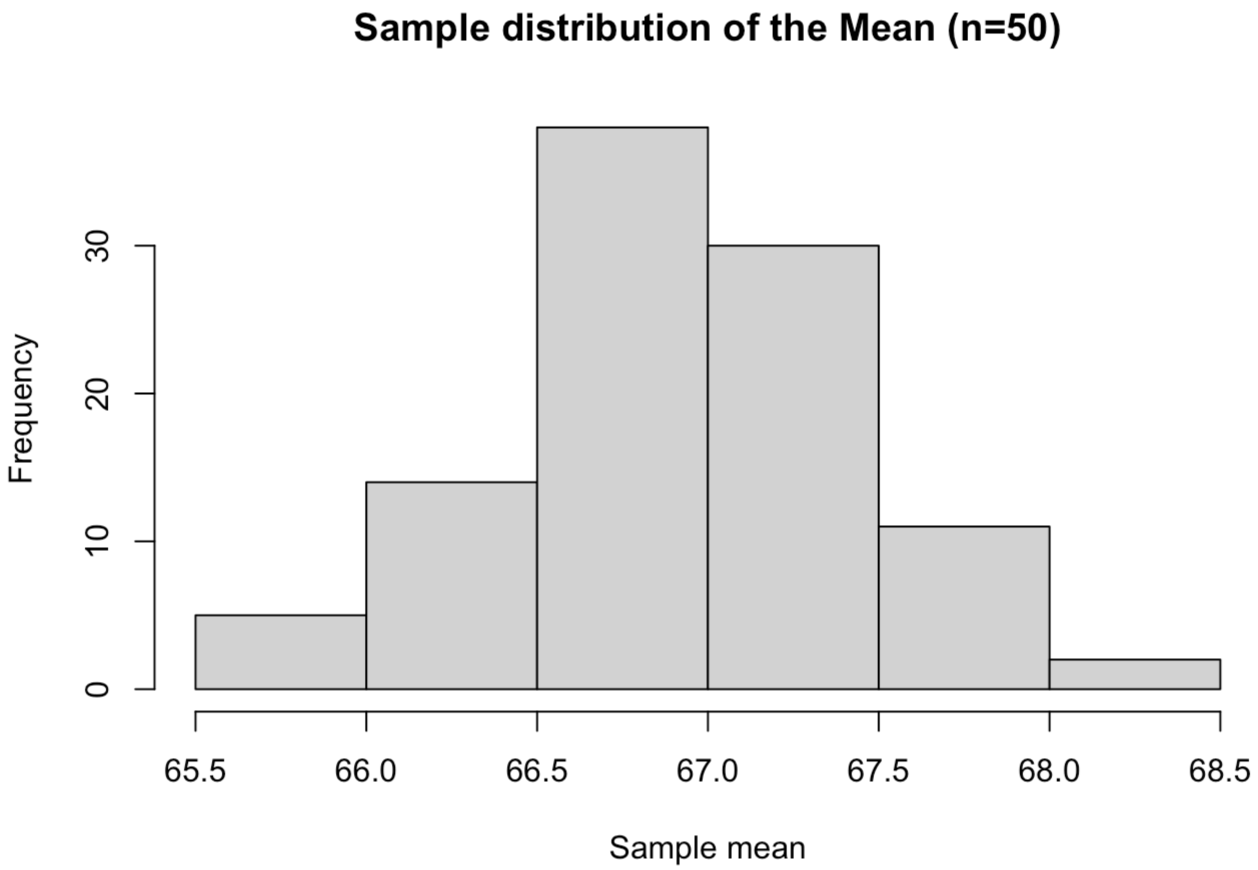
3.2 The observed coverage should be close to 97% (with Monte Carlo fluctuation around it). Expected = 95%.

Q4

```
#parameters
mu <- 67
sd <- 3.5
n <- 50

set.seed(10345)
heights_sample <- sapply(1:100, function(x) rnorm(n, mu, sd) )


#caculate the mean
samples_mean <- colMeans(heights_sample)
hist(samples_mean, main = "Sample distribution of the Mean (n=50)", xlab = "Sample me
an")
```




```
#It likes the normal distribution

mean(samples_mean)
```

```
## [1] 66.92465
```

```
sd(samples_mean)
```

```
## [1] 0.5065171
```

```
sample_se <- apply(heights_sample, 2, function(x) sd(x)/sqrt(n))
t_crits <- qt(0.975, df = n-1)
t_crits
```

```
## [1] 2.009575
```

```
lower_ci <- samples_mean - t_crits*sample_se
upper_ci <- samples_mean + t_crits*sample_se
covered <- (lower_ci <= mu) & (upper_ci >= mu)
covered
```

```
##   [1]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [13]  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [25]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [37]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
##  [49]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [61]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [73]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
##  [85] FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [97]  TRUE  TRUE  TRUE  TRUE
```

```
coverage_percent <- mean(covered) * 100
coverage_percent
```

```
## [1] 93
```

```
list(
  mean_of_sample_means = mean(samples_mean),
  sd_of_sample_means   = sd(samples_mean),
  theoretical_sd       = sd / sqrt(n),
  coverage_percent     = coverage_percent
)
```

```
## $mean_of_sample_means
## [1] 66.92465
##
## $sd_of_sample_means
## [1] 0.5065171
##
## $theoretical_sd
## [1] 0.4949747
##
## $coverage_percent
## [1] 93
```

4.1 Still roughly normal and centered near 67, but wider than for n=100.

4.2 Mean ≈ 67 (expected = 67). SD ≈ 0.495; theoretical SD = 0.495.

4.3 Coverage about 93%; intervals are wider than for n=100 because SE is larger.

Q5

5.1 With n=100, sampling distribution of X is tighter and CIs are narrower because $SE = \sigma/\sqrt{n}$ is smaller (0.35 vs 0.495).

5.2 The histogram becomes smoother and the estimated coverage and summaries are closer to their theoretical values (smaller Monte Carlo error).

5.3 Not exactly. Random samples produce different numerical results each run, but they should follow the same patterns (center near 67, SD near $\sigma/\sqrt{n}$, coverage ≈95%).