HW4.Zunqiu.Wang

Zunqiu Wang

9/23/2021

Q1 a

$$z = \frac{x-\mu}{\sigma}$$

```r
z.score <- function(x, mu, sd) {
  z <- (x-mu) / sd
  return(z)
}
z.score(45, 70, 10)
```

```
## [1] -2.5
```

Q1 b

```r
pnorm(45, 70, 10)
```

```
## [1] 0.006209665
```

Q1 c

```r
cumlfun <- function(x){pnorm(x,70,10)}
lower.prob <- cumlfun(45) # lower tail probability of getting 45 or bolow
# since upper tail is same as lower probability
cumlfun(45)*2 #this is the prob total in both direction by using defination of
```

```
## [1] 0.01241933
```

```r
# following is more comprehensive way of getting score by using definition of z score
score <- function(z, mu, sd) {
  x <- z * sd + mu
  return(x)
}
# z score here is the opposite direction of that of Q1 a since magnitude should be same
score(2.5, 70, 10) # get the score of same magnitude far away from mean
```

```
## [1] 95
```

```r
upper.prob <- 1-cumlfun(95)
lower.prob + upper.prob
```

```
## [1] 0.01241933
```

Q2 a

```r
set.seed(1)
vec <- rpois(10000, 10)
# ggplot(data=data.frame(vec), aes(x=vec)) +  geom_histogram(aes(y=..density..),binwidth=1) +
# xlim(0, 20) + ylab("density") + xlab("outcome")
set.seed(2)
sampl <- sample(vec, 9, replace = TRUE)
```

Q2 b

$$mean = \frac{1}{n}\sum_{i=i}^{n} x_i$$

```r
sample.mean <- (11+10+12+11+11+10+10+10+7)/9
sample.mean
```

```
## [1] 10.22222
```

```r
mean(sampl)
```

```
## [1] 10.22222
```

Q2 c

$$sd = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

```r
n=9
sample.sd <- sqrt((1/(n-1))*((11-sample.mean)^2 * 3
                            + (10-sample.mean)^2 * 4 + (7-sample.mean)^2 + (12-sample.mean)^2))
sd(sampl)
```

```
## [1] 1.394433
```

Q2 d

$$se = \frac{s}{\sqrt{n}}$$

```r
se <- sample.sd/sqrt(9)
```

Q2 e

$$CI = \bar{x} \pm z * se$$

$$p(\bar{x} - 1.959964se \ < \ \mu \ < \bar{x} + 1.95964se) = 0.95$$

$$p(10.22222 - 1.959964 * 0.4648111 < \mu < 10.22222 + 1.959964 * 0.4648111) = 0.95$$

$$p(9.311207 < \mu < 11.13323) = 0.95$$

```r
## z score for 95% CI
z.score.95 <- qnorm(0.975)
10.22222 - z.score.95 * se
```

```
## [1] 9.311207
```

```r
10.22222 + z.score.95 * se
```

```
## [1] 11.13323
```

Q2 f

$$p(\bar{x} - T(0.975, 8) * se < \mu < \bar{x} + T(0.975, 8) * se) = 0.95$$

$$p(10.22222 - 2.306004 * 0.4648111 < \mu < 10.22222 + 2.306004 * 0.4648111) = 0.95$$

$$p(9.150364 < \mu < 11.29408) = 0.95$$

```r
# get t value for 95% CI with df=8
t.val <- qt(0.975, 8)
10.22222-2.306004*0.4648111
```

```
## [1] 9.150364
```

```r
10.22222+2.306004*0.4648111
```

```
## [1] 11.29408
```

Q3 a - the sample size is 9, which is smaller than 30 so using Z distribution is inappropriate but using t distribution

Q3 b - using standard deviation instead of standard error, t value is incorrectly chosen ($t_{0.1}$ with df=4) but should be ($t_{0.05}$ with df=n-1=3)
- t value is incorrectly chosen ($t_{0.1}$ with df=4) but should be ($t_{0.05}$ with df=n-1=3)
- using standard deviation instead of standard error, t value is incorrectly chosen ($t_{0.1}$ with df=3) but should be ($t_{0.05}$ with df=3)
- correct
- t value is incorrectly chosen ($t_{0.05}$ with df=4) but should be ($t_{0.05}$ with df=3)

Q4 a

```r
qt(0.975, n-1) * (sample.sd / sqrt(n))
```

```
## [1] 1.071856
```

```r
total.ppl <- (1.394433*2.306004/0.535928)^2
```

- assuming

  $$\bar{x}\ and\ s$$

  don't change with addition of individuals in the sample

- intended interval: 1/2 * ((10.22222 + 2.306004 * 0.4648111) - (10.22222 - 2.306004 * 0.4648111)) = 1.071856 1.071856/2=0.535928

$$0.535928 = 2.306004 * \frac{1.394433}{\sqrt{n}}$$
$$n = 36$$
$$36 - 9 = 27$$

- additional 27 individuals are needed

Q4 b

$$CI = z * \frac{s}{\sqrt{n}}$$

```
# assume s and mean dont change with additional individuals for the sample under normal distribution
# to get z score
qnorm(0.975)
```

```
## [1] 1.959964
```

```
# get n
(1.959964*20000/1000)^2
```

```
## [1] 1536.584
```

```
(1.959964*20000/100)^2
```

```
## [1] 153658.4
```

$$1000 = 1.959964 * \frac{20000}{\sqrt{n}}$$
$$n = 1536$$
$$100 = 1.959964 * \frac{20000}{\sqrt{n}}$$
$$n = 153658$$

Q5

```
set.seed(1234)
# 1. set sample size
nsamples <- 20
# 2. set how many times running the whole thing
nruns <- 1000
# 3. create empty matrix to store summary stats
sample.summary <- matrix(NA, nruns, 3)
# 4. outer loop
for (j in 1:nruns) {
  sampler <- rep(NA, nsamples)
  # 5. run inner sampling loop to construct a normal distribution
  for (i in 1:nsamples) {
```

```r
    sampler[i] <- rnorm(1, 30, 4)
  }
  # 6. calculate summary stats:mean, 99% CI and save into matrix
  sample.summary[j, 1] <- mean(sampler) # mean
  se <- sd(sampler)/sqrt(nsamples) # se
  sample.summary[j, 2] <- mean(sampler) - qt(0.997, length(sampler) - 1) * se # lower 99% CI bound
  sample.summary[j, 3] <- mean(sampler) + qt(0.997, length(sampler) - 1) * se # upper 99% CI bound
}
counter = 0
for (j in 1:nruns) {
  # if mean is 99% within theoretical 99% CI bound
  if (30 > sample.summary[j, 2] && 30 < sample.summary[j, 3]) {
    counter <- counter + 1
  }
}
counter
```

```
## [1] 992
```

```r
counter/nruns
```

```
## [1] 0.992
```

```r
# it turns out that the asuumption that sample mean are distributed
# normally with se is a good referance. It validates the theoretical
# 99% CI is right 99% of the time. According CI formula, when sample
# size is small the CI will be larger and leads to increased uncertainty
# about true mean but it still supports the claim that true mean is
# 99% of time in the CI bound.
```