Confidence Intervals

22 marks

In this question, the **meaning** of a confidence interval will be explored by simulation.

Suppose we have the model that Y_1, Y_2, ..., Y_n are independently and identically distributed $N(\mu, \sigma^2)$ and we wish to construct a 95% confidence interval for the unknown **parameter** μ from a sample of realized values y_1, y_2, ..., y_n.

The standard **estimates** of μ and σ are

$$\hat{\mu} = \overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

and

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{\mu})^2}{n-1}} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \overline{y})^2}{n-1}}$$

with corresponding **estimators**

$$\tilde{\mu} = \overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

and

$$\tilde{\sigma} = \sqrt{\frac{\sum_{i=1}^{n}(Y_i - \tilde{\mu})^2}{n-1}} = \sqrt{\frac{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}{n-1}}$$

The interval

$$\left[\hat{\mu} - c\frac{\hat{\sigma}}{\sqrt{n}}\ ,\ \hat{\mu} + c\frac{\hat{\sigma}}{\sqrt{n}}\right]$$

where c is the constant such that for the Student t random variable on $n-1$ degrees of freedom

$$Pr(t_{n-1} \leq c) = 1 - \frac{\alpha}{2} \quad \text{or equivalently} \quad Pr(|t_{n-1}| \leq c) = 1 - \alpha.$$

That is, $c = Q_{t_{n-1}}(1 - \frac{\alpha}{2})$ is the $p = 1 - \frac{\alpha}{2}$ quantile of a t_{n-1} random variable.

This interval is called a $100(1-\alpha)\%$ **confidence** interval because its random counterpart

$$\left[\tilde{\mu} - c \times \widetilde{SD}(\tilde{\mu})\ ,\ \tilde{\mu} + c \times \widetilde{SD}(\tilde{\mu})\right] = \left[\tilde{\mu} - c\frac{\tilde{\sigma}}{\sqrt{n}}\ ,\ \tilde{\mu} + c\frac{\tilde{\sigma}}{\sqrt{n}}\right]$$

will contain (or cover) the true value μ with probability $(1-\alpha)$. This is called its **coverage probability**.

In this question, you are going to generate many random intervals and observe their empirical coverage.

Before getting started, there is an R function `t.test()` that will be of some value (See `help(t.test)`.) For example,

- `t.test(y, conf.level = 0.95)$conf.int`

  returns the 95% confidence interval for μ based on the vector `y` of realizations $y_1, \ldots, y_n$ and
- `t.test(y, mu = a)$p.value`

  returns the p-value for testing the hypothesis $H_0 : \mu = a$ against the "two sided" alternative $H_a : \mu \neq a$.

Answer each of the following questions **showing your code for every part**.

a. **(4 marks)** Complete the following function

```
conf.intervals <- function(mu = 0,           # true mean of normals
                           sigma = 1,        # true sd of normals
                           sampleSize = 100, # size of normal sample
                           level = 0.95,     # confidence level
                           nIntervals = 20   # number of intervals
                           ){
                           ...
                           }
```

which returns a `data.frame` having `nIntervals` rows and two columns with variable names `lwr` and `upr` representing the lower and upper values of a \$100 ×level`$\%$ confidence interval for $\mu$ based on a sample of size`sampleSize`from a normal distribution with mean`mu`and standard deviation`sigma`‘.

Each row is a single confidence interval for on a different independent normal sample of size `sampleSize`.

- show your code
- show the output of your function by evaluating

```
conf.intervals <- function(mu = 0,           # true mean of normals
                           sigma = 1,        # true sd of normals
                           sampleSize = 100, # size of normal sample
                           level = 0.95,     # confidence level
                           nIntervals = 20   # number of intervals
                           ){
   conf_df <- data.frame()
   for(i in 1:nIntervals){
      data <- rnorm(sampleSize, mu, sigma)
      t_stats <- t.test(data, conf.level = level) #contains statistics, parameter, p.value, conf.int, .
      conf_interval <- t_stats$conf.int  #gives us confidence interval: lwr & upr
      conf_df[i,'lwr'] <- conf_interval[1]
      conf_df[i,'upr'] <- conf_interval[2]
   }
   return(conf_df)
}
set.seed(1234567)
head(conf.intervals(), 2)
```

```
##          lwr       upr
## 1 -0.2673282 0.1130814
## 2 -0.2107598 0.1944589
```

b. **(2 marks)** Use the function you defined in (a) to construct a dataset of 100 90 confidence intervals for μ from samples of size $n = 30$ from $N(\mu, \sigma^2)$ with $\mu = 10$ and $\sigma = 3$.

Assign this data set to the variable `intervals90` as in

and demonstrate the values using

```
#set.seed(1234567)
intervals90 <- conf.intervals( mu = 10,           # true mean of normals
                          sigma = 3,         # true sd of normals
                          sampleSize = 30, # size of normal sample
                          level = 0.90,      # confidence level
                          nIntervals = 100)
head(intervals90, 2)
```

```
##        lwr      upr
## 1 9.025931 10.77421
## 2 9.013557 10.96368
```

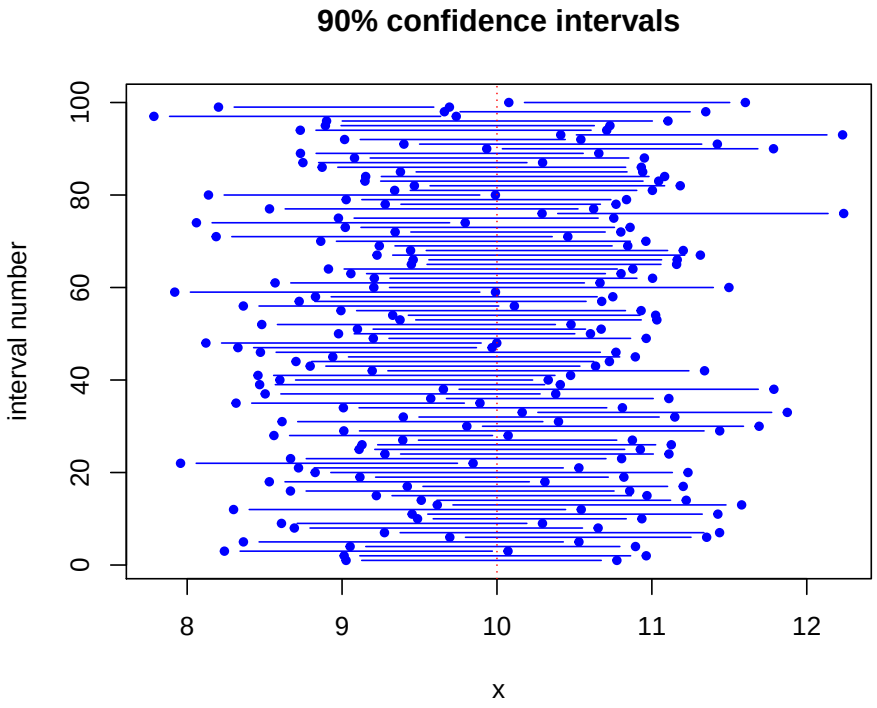
c. **(3 marks)** For the confidence intervals in `intervals90`, construct a plot showing each interval as a horizontal line segment showing the location of that interval. For the ith confidence interval the line segment should have horizontal (x) values corresponding to its lower and upper values and vertical (y) values equal to i.

Add a single **red** vertical line at the true value of μ.

Hint: After defining the horizontal limits `xlim` and vertical limits `ylim` of the plot, you can begin plotting with an "empty" plot defined as follows

```
plot(0, type = "n", xlim = xlim, ylim = ylim,
     xlab = "x", ylab = "interval number",
     main = "90% confidence intervals")
```

```
lowestX <- min(intervals90$lwr)
highestX <- max(intervals90$upr)
xlim <- c(lowestX, highestX)
ylim <- c(1,100)
plot(0, type = "n", xlim = xlim, ylim = ylim,
     xlab = "x", ylab = "interval number",
     main = "90% confidence intervals")
for(i in 1:100) {lines(c(intervals90$lwr[i],intervals90$upr[i]), c(i,i), pch = 20, col = "blue", type =
abline(v=10, col="red", lty=3)
```



The line segments for the confidence intervals can be added using `lines()` one at a time in a loop as for example `for(i in 1:100) {lines(...)}`

d. **(3 marks)** Write some code that counts the number of intervals in `intervals90` that cover the true value of μ.

```
count <- 0
for (i in 1:100){
   if ((intervals90$lwr[i] < 10) & (intervals90$upr[i] > 10)){
      count <- count + 1
   }
}
count
```

```
## [1] 87
```

- how many cover μ? Ans: 87
- does this makes sense? Explain your reasoning.

The count makes sense because in intervals90, 87 covers the μ and 13 don't. It makes sense because 87 is not much different from 90.

e. **(3 marks)** If x is the number of $100(1-\alpha)\%$ intervals covering the true value μ out of m independently generated intervals, what is the probability distribution of X?

That is, what is

$$Pr(X = x) = ?$$

Explain your reasoning.

$$Pr(X = x) = \binom{n}{k} p^k (1-p)^{n-k}$$

The reason why the probability distribution of X is binomial distribution is because there are only two possible outcomes. The two outcomes are 1. intervals covering the true valueμ 2.intervals not covering the true value μ

f. **(3 marks)** Complete the following function

```
p.values <- function(mu = 0,           # true mean of normals
                     sigma = 1,        # true sd of normals
                     sampleSize = 100, # size of normal sample
                     mu_0 = 0,         # hypothesized mean
                     nSamples = 20     # number of samples
                           ){
                           ...
                           }
```

which returns a `vector` of `nSamples` p-values for testing the hypothesis $H_0 : \mu = \mu_0$ against the "two sided" alternative $H_a : \mu \neq \mu_0$ based on `nSamples` independent samples of size `sampleSize` from a normal distribution with true mean `mu` and true standard deviation `sigma`.

Each element of the vector returned is a single p-value testing $H_0 : \mu = \mu_0$ against the "two sided" alternative $H_a : \mu \neq \mu_0$ based on a different independent normal sample of size `sampleSize`. There will be `nSamples` elements.

- show your code
- show the output of your function by evaluating

```
set.seed(1234567)
p.values <- function(mu = 0,         # true mean of normals
                     sigma = 1,      # true sd of normals
                     sampleSize = 100, # size of normal sample
                     mu_0 = 0,       # hypothesized mean
                     nSamples = 20   # number of samples
                     ){
   v <- vector(mode="numeric", length=nSamples)
   test <- data.frame()
   for(i in 1:nSamples){
      test <- rnorm(sampleSize, mu, sigma)
      stats <- t.test(test,mu = mu_0)
      v[i] <- stats$p.value
   }
   return(v)
}
head(p.values(), 2)
```

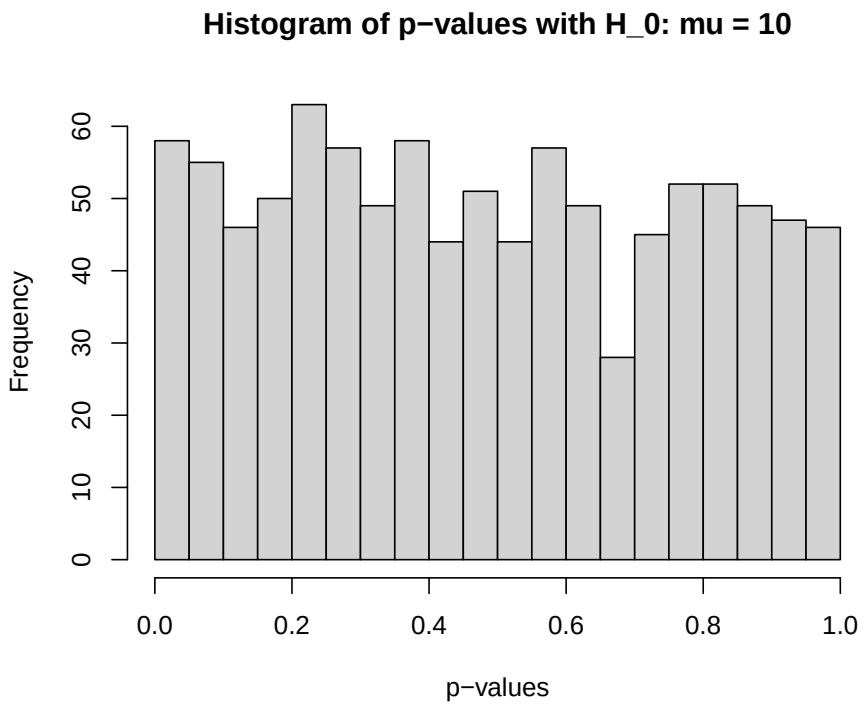
```
## [1] 0.4230064 0.9365414
```

g. **(3 marks)** Get the p-values for 1000 samples, each of size 50, drawn from $N(10, 9)$ where on each sample the hypothesis $H_0 : \mu = 10$ is tested against the two-sided alternative. Save the result as the variable `pvals`.

Draw a histogram of the `pvals` you just constructed.

- describe the distribution
- does this make sense? Why? Or Why not?

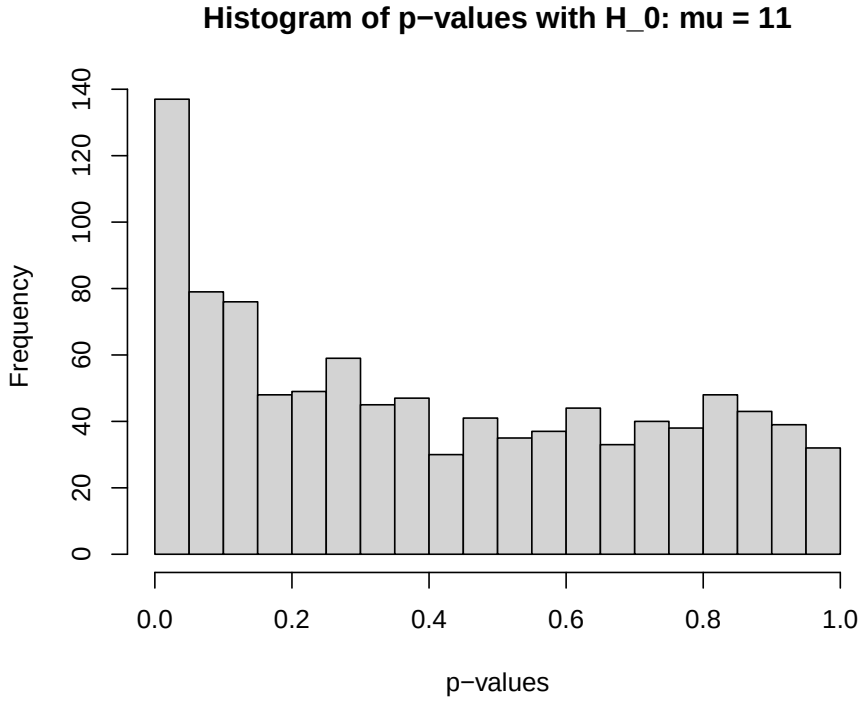
```
pvals <- p.values(10, 9, 50, 10, 1000)
hist(pvals, breaks = 30, xlab = "p-values", main = "Histogram of p-values with H_0: mu = 10")
```



The distribution looks like a uniform distribution. From the histogram of p-values with $\mu = 10$, we can see that the frequencies are almost the same for each p-values. This makes sense because the Null hypothesis is μ =10.

h. **(4 marks)** Repeat part (g) but this time test the hypothesis $H_0 : \mu = 11$. In addition, what do you imagine this histogram would look like if the size of each sample was $n = 100$ instead of $n = 50$.

```
pvals <- p.values(11, 9, 50, 10, 1000)
hist(pvals, breaks = 30, xlab = "p-values", main = "Histogram of p-values with H_0: mu = 11")
```



The distribution looks like it is positively skewed. From the histogram of p-values with $\mu = 11$, we can see that the frequencies decreases as p-values increases. And the frequency is highest when p-values $= 0$, this makes sense because the null hypothesis we are testing now is $\mu = 11$.