Applied Statistics 2021 - Exercise 11

1. Bags of potatoes (Theory)

You bought 10 very large bags of potatoes. Assume that the 10 weights can be viewed as a realization of a random sample from a normal distribution with unknwown parameters. Your measures give you the following data:

- Sample mean: 14.5 kg
- Sample standard deviation: 0.3 kg

Construct a 95% confidence interval for the expected weight of a bag.

Solution:

We model the data using the Student's t-distribution with $\alpha = 0.05$ to find the confidence interval (Dekking p. 349 + 350), as we don't know the true standard deviation of the potato distribution.

The 95% confidence interval for μ of the bag weights is then

$$\begin{aligned} &\left(\bar{x}_n - t_{n-1,\alpha/2}\frac{s_n}{\sqrt{n}}, \bar{x}_n + t_{n-1,\alpha/2}\frac{s_n}{\sqrt{n}}\right) \\ =&\left(14.5 - t_{9,0.025}\frac{0.3}{\sqrt{10}}, 14.5 + t_{9,0.025}\frac{0.3}{\sqrt{10}}\right) \end{aligned}$$

Let's use R to find $t_{9,0.025}$:

```
qt(0.025,9,lower.tail = F)
```

```
## [1] 2.262157
```

We plug that in to find the 95% confidence interval

$$\begin{aligned} &\left(14.5 - 2.262\frac{0.3}{\sqrt{10}}, 14.5 + 2.262\frac{0.3}{\sqrt{10}}\right) \\ =&(14.285, 14.715) \end{aligned}$$

2. How many samples do we need? (Theory)

Assume that we measure a person's height (in meters) and that the measurements are normal distributed with standard deviation $\sigma = 0.01$. How many measurements do we to make, if we want a 99% confidence interval no wider than 0.001 meters for the mean μ? Please explain how you find the number of required measurements.

Solution:

We know the standard deviation, and we can therefore model their heights with a normal distribution (and not a Student's t-distribution):

$$\left(\bar{x}_n - z_{\alpha/2}\frac{\sigma}{\sqrt{n}}, \bar{x}_n + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right)$$

where $z_{\alpha/2}$ is defined as $P(Z \geq z_{\alpha/2}) = \alpha/2$ for $Z \sim N(0,1)$.

The width of this confidence interval is

$$2 \cdot z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$

We are given all the information except n. We have $\alpha = 1 - (99/100) = 0.01$, the width (w) is 0.001 and $\sigma = 0.01$. Taking the aforementioned equation:

$$w = 2 \cdot z_{\alpha/2} \frac{\sigma}{\sqrt{n}}$$
$$\sqrt{n} = \frac{2 \cdot z_{\alpha/2} \cdot \sigma}{w}$$
$$n = \left(\frac{2 \cdot z_{\alpha/2} \cdot \sigma}{w} \right)^2$$

```
alpha <- 0.01
sigma <- 0.01
width <- 0.001
z <- qnorm(alpha / 2, lower.tail = FALSE)
n <- ceiling((2 * z * sigma / width)^2)
sprintf("We need to take at least %d measurements to meet the given requirements", n)
```

```
## [1] "We need to take at least 2654 measurements to meet the given requirements"
```

3. Tomato sauce (Theory)

A team of scientists is analyzing samples from bottles of tomato sauce, looking for traces of pesticides. They take a random sample of 10 bottles, with the assumption is that the samples are drawn from a normal distribution with unknown mean and standard deviation. The scientsts find that the sample mean is $\bar{x}_{10} = 1.05655721956321$ parts per million (ppm), with a sample standard deviation $s_{10} = 0.200131346424344$. The maximum limit prescribed by the law is 1.15 ppm.

a. The team of scientists wants to derive a *one sided* 90% confidence interval for the mean concentration of pesticides in the bottles, because they want to sue the tomato brand for having too many pesticides in their product. Which type of one sided interval should they use? Derive such an interval.

Solution:

We want a lower bound for the pesticide concentration. We use a t-distribution as we don't have access to the true σ of the pesticide concentration distribution, and so we want a confidence interval of the type (Dekking p. 366):

$$\left(\bar{x}_n - t_{n-1,\alpha} \frac{s_n}{\sqrt{n}}, \infty \right)$$

with $\alpha = 1 - (90/100)$, $S_n = 0.200$, $\bar{X}_n = 1.057$ and $n = 10$.

Let's use R to find the confidence interval

```
alpha = 1-90/100
sample.std = 0.200131346424344
sample.mean = 1.05655721956321
n = 10
lower.bound = sample.mean - qt(alpha, n - 1, lower.tail = F)*sample.std / sqrt(n)
print(sprintf("The 90%% confidence interval for the mean is (%.3f, inf)",lower.bound))
```

```
## [1] "The 90% confidence interval for the mean is (0.969, inf)"
```

b. The tomato brand defends itself by deriving, from the very same data, another one sided 90% confidence interval. Which one? Compute it.

Solution:

The tomato brand wants a 90% upper bound on the mean of the pesticide concentration. Very similar procedure:

$$\left(0, \bar{x}_n + t_{n-1,\alpha}\frac{s_n}{\sqrt{n}}\right)$$

```
alpha = 1-90/100
sample.std = 0.200131346424344
sample.mean = 1.05655721956321
n = 10
upper.bound = sample.mean + qt(alpha, n - 1, lower.tail = F)* sample.std / sqrt(n)
print(sprintf("The 90%% confidence interval for the mean is (0, %.3f)",upper.bound))
```

```
## [1] "The 90% confidence interval for the mean is (0, 1.144)"
```

c. Which conclusions can be derived from these intervals?

Solution:

Since the maximum allowed concentration is 1.15 ppm and we have a 90% upper bound on the concentration of 1.144 ppm, the tomato brand seems to comply with the law.

d. The scientists are not yet convinced, therefore they perform a second experiment. This time they take a random sample of 100 bottles, finding the sample mean $\bar{x}_{100} = 1.14376626794198$ and the sample standard deviation $s_{100} = 0.173021605092371$. Recompute the confidence intervals of points a. and b. Is the tomato brand still able to defend itself?

Solution:

```
alpha = 1-90/100
sample.std = 0.173021605092371
sample.mean = 1.14376626794198
n = 100
lower.bound = sample.mean - qt(alpha, n - 1, lower.tail = F)*sample.std / sqrt(n)
upper.bound = sample.mean + qt(alpha, n - 1, lower.tail = F)* sample.std / sqrt(n)
print(sprintf("The 90%% confidence interval for the mean is (%.3f, inf)",lower.bound))
```

```
## [1] "The 90% confidence interval for the mean is (1.121, inf)"
```

```
print(sprintf("The 90%% confidence interval for the mean is (0, %.3f)",upper.bound))
```

```
## [1] "The 90% confidence interval for the mean is (0, 1.166)"
```

Now the upper 90% bound on the concentration is 1.166, which is above the maximum allowed concentration of 1.15. It doesn't look good for the tomato company.

4. Simulating confidence intervals (R)

In this first exercise you will build confidence intervals from different samples of a normal random variable.

a. Define a function `confidence_interval_normal` that accepts two parameters: a sample of elements and a confidence level. Under the assumption of normality of the input data, build the confidence interval using the formula for normal variable with unknown mean and variance. The function should return the interval as a two element vector: `c(lower, upper)`. To get the quantiles of the t-distribution

you can use the `qt` function: `qt(0.95, 19)` return the 0.95 quantile of the t-distribution with 19 degrees of liberty. See `help(qt)` for more details.

Solution:

```
confidence_interval_normal <- function(sample.elements, confidence.level){
  n = length(sample.elements)
  alpha = 1 - confidence.level
  sample.mean = mean(sample.elements)
  sample.std = sd(sample.elements)
  return(c(sample.mean - qt(alpha/2, n-1, lower.tail = F)*sample.std / sqrt(n),
           sample.mean + qt(alpha/2, n-1, lower.tail = F)*sample.std / sqrt(n)))
}
```

b. Perform the following simulation: run 100 iterations, in each iteration take a sample of 10 elements using `rnorm(10, 150, 50)`. For each such sample, use the function developed in the previous point to compute a 0.9 confidence interval relative to the data. Count the number of intervals that *do not* contain the true mean of the distribution we are sampling from, that is `150`. Is the result in line with your expectations?

Solution:

```
reps = 100

n = 10
mu = 150
std = 50


in.interval = 0
for (i in 1:reps){
  X = rnorm(n,mu,std)
  confidence.interval = confidence_interval_normal(X,0.9)
  if (mu>=confidence.interval[1] && mu<=confidence.interval[2])
    in.interval = in.interval + 1
}
print(sprintf("%d of the runs had the true mu in the interval",in.interval))
```

```
## [1] "92 of the runs had the true mu in the interval"
```

```
print(sprintf("%d of the runs did not have the true mu in the interval",reps - in.interval))
```

```
## [1] "8 of the runs did not have the true mu in the interval"
```

This result is in line with my expectations.

c. Try to change the confidence level, setting it to 0.8 and 0.95. How does the result change?

Solution:

```
in.interval.0.8 = 0
in.interval.0.95 = 0
for (i in 1:reps){
  X = rnorm(n,mu,std)
  confidence.interval.0.8 = confidence_interval_normal(X,0.8)
  confidence.interval.0.95 = confidence_interval_normal(X,0.95)
  if (mu>=confidence.interval.0.8[1] && mu<=confidence.interval.0.8[2])
    in.interval.0.8 = in.interval.0.8 + 1
  if (mu>=confidence.interval.0.95[1] && mu<=confidence.interval.0.95[2])
```

```
    in.interval.0.95 = in.interval.0.95 + 1
}
print(sprintf("%d of the runs had the true mu in the 80%% interval",in.interval.0.8))
```

```
## [1] "79 of the runs had the true mu in the 80% interval"
```

```
print(sprintf("%d of the runs did not have the true mu in the 80%% interval",reps - in.interval.0.8))
```

```
## [1] "21 of the runs did not have the true mu in the 80% interval"
```

```
print(sprintf("%d of the runs had the true mu in the 95%% interval",in.interval.0.95))
```

```
## [1] "96 of the runs had the true mu in the 95% interval"
```

```
print(sprintf("%d of the runs did not have the true mu in the 95%% interval",reps - in.interval.0.95))
```

```
## [1] "4 of the runs did not have the true mu in the 95% interval"
```

5. Cats & confidence (R)

The `cats` data set (available in `UsingR`) contains the bodyweight and heart weight of adult cats, along with their sex.

(a) Compute the mean and the 90% confidence intervals for the body weight and heart weight assuming normality of the samples. Do the computation separately for the female and male cats. (So you have to compute 4 confidence intervals). Is there any difference between the results? You can use the `confidence_interval_normal` function that you defined in the previous exercise.

Solution:

```
library(UsingR)
```

```
## Loading required package: MASS
## Loading required package: HistData
## Loading required package: Hmisc
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
## Loading required package: ggplot2
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
##
## Attaching package: 'UsingR'
## The following object is masked from 'package:survival':
##
##     cancer
```

```r
body.male <- cats$Bwt[cats$Sex == "M"]
heart.male <- cats$Hwt[cats$Sex == "M"]
body.female <- cats$Bwt[cats$Sex == "F"]
heart.female <- cats$Hwt[cats$Sex == "F"]



body.male.conf = confidence_interval_normal(body.male,0.9)
heart.male.conf = confidence_interval_normal(heart.male,0.9)
body.female.conf = confidence_interval_normal(body.female,0.9)
heart.female.conf = confidence_interval_normal(heart.female,0.9)

print(sprintf("90%% confidence interval for the body weight of male cats: (%.3fkg, %.3fkg)",
              body.male.conf[1], body.male.conf[2]))
```

```
## [1] "90% confidence interval for the body weight of male cats: (2.821kg, 2.979kg)"
```

```r
print(sprintf("90%% confidence interval for the heart weight of male cats: (%.3fg, %.3fg)",
              heart.male.conf[1], heart.male.conf[2]))
```

```
## [1] "90% confidence interval for the heart weight of male cats: (10.894g, 11.751g)"
```

```r
print(sprintf("90%% confidence interval for the body weight of female cats: (%.3fkg, %.3fkg)",
              body.female.conf[1], body.female.conf[2]))
```

```
## [1] "90% confidence interval for the body weight of female cats: (2.292kg, 2.427kg)"
```

```r
print(sprintf("90%% confidence interval for the heart weight of female cats: (%.3fg, %.3fg)",
              heart.female.conf[1], heart.female.conf[2]))
```

```
## [1] "90% confidence interval for the heart weight of female cats: (8.870g, 9.535g)"
```

(b) Compute the one-sided 95% confidence intervals for the mean body weight of male cats and compare to the results you obtained at (a).

Solution:

```r
body.male <- cats$Bwt[cats$Sex == "M"]

male.body.mean = mean(body.male)
male.body.std = sd(body.male)
n = length(body.male)
alpha = 1 - 95/100

lower.bound = male.body.mean - qt(alpha, n - 1, lower.tail = F)*male.body.std / sqrt(n)
upper.bound = male.body.mean + qt(alpha, n - 1, lower.tail = F)* male.body.std / sqrt(n)

print(sprintf("95%% confidence interval for male body weight: (%.3fkg, inf kg)",lower.bound))
```

```
## [1] "95% confidence interval for male body weight: (2.821kg, inf kg)"
```

```r
print(sprintf("95%% confidence interval for male body weight: (0 kg, %.3fkg)",upper.bound))
```

```
## [1] "95% confidence interval for male body weight: (0 kg, 2.979kg)"
```

6. Bootstrapping confidence intervals (R)

Consider again the `cats` dataset of the previous exercise. Construct the 90% confidence intervals for the mean body weight of male cats by empirical bootstrap, using 500 bootstrap repetitions. Compare the result to those you got in Problem 1.
How do they compare with the intervals you found in the previous exercise?

Solution:

We use the method from p. 351 of Dekking. A 90% confidence interval corresponds to the quantiles 0.05 and 0.95 of the empirical distribution function:

```
bootstrap.means = c()
for (i in 1:500)
  bootstrap.means = c(bootstrap.means, mean(sample(body.male, length(body.male), replace = T)))

quantile(bootstrap.means, c(0.05, 0.95))
```

```
##       5%      95%
## 2.822629 2.983505
```