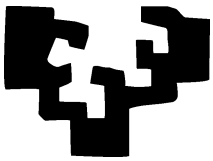
eman ta zabal zazu



Universidad del País Vasco

Euskal Herriko Unibertsitatea

Activity 1

1 Synopsis.

Quite often in Statistics we attempt to learn something about a *population* which we cannot fully inspect on the basis of what we observe in a subset (a *sample*) taken from it[1].

For instance, we may face a population with unknown variance whose value we want to approximate or *estimate*. Such approximation is usually computed from the values of a sample.

We can estimate a variance, or any other *parameter* of a population, in virtually endless ways, some better than others. Usually, but not always, statistical theory will guide us in the choice of good methods of estimation. If everything else fails, though, we can always resort to the Monte Carlo method presented in Seminar 1.

What you need to know. In order to benefit from this activity you only need to know some rudiments of R. The introduction in Seminar 1 may well be enough. Review your notes and remember the use of instructions such as `runif`, `rnorm`, `if` and `for`.

2 The problem

2.1 Two estimates of the variance

By now you are probably familiar with two different ways to estimate the variance $\sigma^2 = E[(X-m)^2]$ of a population. Given a sample of size n from it, we can either compute

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2 \tag{1}$$

[1] More on *sampling* as well as a fuller explanation of all the italiced words that follow will occupy us the second half of the course.

or

$$s_*^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2. \tag{2}$$

You may surmise that the difference between the two estimates cannot be that much, particularly if n is any big. Still, it is interesting to investigate exactly what is the difference and which one is better, according to different criteria of "betterness". Do as instructed in Section 2.2 next.

2.2 Work to do

1. For sample sizes $n = 10, 100$ and 1000 generate $N = 1000$ samples from a normal distribution $N(0, \sigma^2 = 1)$.

2. Each time you generate a random sample, compute both (1) and (2) and save the values. At the end, you should have two matrices (call them `var1` and `var2`) with three columns each containing one thousand estimates obtained from that many samples. This is quite easy, it will suffice to use two nested loops like:

```
> N     <- 1000
> sizes <- c(10,100,1000)
> var1 <- var2 <- matrix(0,N,3)
> for (i in 1:3) {
      n <- size[i]
      for (j in 1:N) {

         #  do whatever is needed here

      }
  }
```

   Notice that you are here in the (unrealistic) situation in which you *know* what the value of the parameter σ^2 is. This affords you the luxury of being able to assess each possible estimator by comparison with the (usually unavailable) true value.

3. Now, for each sample size and each of the two competing estimators (1) and (2) you have one thousand realizations. Except by sheer luck, not one will be exactly equal to the true, known value of σ^2, but either of the two estimators may appear to approach the target better or worse than the other. Here are a few suggestions you may try: check your results and communicate your findings.

   (a) Which of the two estimators appears *on average* to be closer to the true value of 1? To answer that for e.g. the estimator based on samples of size $n = 10$ (first column of the matrices), you might compute

```
> mean(var1[,1]) - 1
> mean(var2[,1]) - 1
```

and likewise for the second and third columns[2].

(b) Is that average deviation from the true value a good indication of "goodness" of the estimator? (Hint: It might happen that one estimator always misses badly the target, sometimes being much larger, sometimes much smaller than the true value, yet *on average* is about right.)

(c) *(connected with previous question)* Consider now a different way of judging estimators. We will judge an estimator good using the average of squared deviations from the true value i.e. for the first estimator (`var1`) and first sample size (column 1),

```
> mean( (var1[,1] - 1)^2 )
```

Clearly, we would like this figure to be small. Compute it for the two estimators and three sample sizes and draw your conclusions[3].

(d) What happens with

```
> mean(var1[,1]) - 1
> mean(var2[,1]) - 1
```

when you consider larger sample sizes (that is, you take the second or third column of `var1` and `var2`)?

References

[2]The differences you are computing here are similar to what we will call *bias* later in the course.

[3]The average square differences that we are computed here can be seen as approximations to what we will later call *mean square error* (MSE).

3 Posibles respuestas

Pueden completar la ayuda que se les da escribiendo código como:

```
> N     <- 1000
> sizes <- c(10,100,1000)
> var1 <- var2 <- matrix(0,N,3)
> colnames(var1) <- colnames(var2) <- c("n10","n100","n1000")
> for (i in 1:3) {
      n <- sizes[i]
      for (j in 1:N) {
          muestra <- rnorm(n,0,1)
          m       <- mean(muestra)
          s2      <- sum( (muestra-m)^2 ) / n
          s2star  <-  sum( (muestra-m)^2 ) / (n-1)
          var1[j,i] <- s2
          var2[j,i] <- s2star
      }
  }
```

Una vez pobladas las dos matrices podemos estimar los sesgos así (se dan formas alternativas de hacerlo; los alumnos utilizarán seguramente un bucle más farragoso de escribir, pero igualmente correcto):

```
> apply( (var1-1), 2, mean)

         n10         n100        n1000
-0.116648582 -0.010655491  0.001391478

> colMeans(var2 - 1)

          n10          n100         n1000
-0.0184984243 -0.0006621119  0.0023938720
```

y los ECM así:

```
> apply( (var1-1)^2, 2, mean)

        n10        n100       n1000
0.184441074 0.019448332 0.001810834

> colMeans( (var2 - 1)^2 )

        n10        n100       n1000
0.211248590 0.019727806 0.001818252
```

El estimador en `var1` es sesgado (por defecto), cosa que es particularmente visible con muestras muy pequeñas ($n = 10$). A medida que n crece, ambos estimadores dan resultados prácticamente idénticos en cuanto a sesgo.

El estimador en `var1`, pese a su sesgo, proporciona sistemáticamente menor ECM que el estimador insesgado en `var2`. De nuevo esto es particularmente apreciable cuando $n = 10$. Aquí tenemos un ejemplo de estimador sesgado que domina (en términos de ECM) a otro insesgado; se lo podremos recordar más adelante en el curso, en la parte de inferencia.