Statistics 251: Lab 6 Pre-Reading – Inference (Part I): Point Estimates

Objectives:

- Understand point estimates
- Review
  - *statistics* are random variables
  - sample variance
- Learn additional R commands including `for loops`
- Prepare for lab 6 by trying a pre-lab exercise

1.0 Introduction

A big reminder on a key concept in this course:

Statistics are Random Variables!

1.1 Point Estimates

A point estimate is a certain combination of the sample measurements (a function of the sample) which is expected to take values "reasonably close" to the parameter it is supposed to estimate. Our aim is to estimate some unknown *parameter* based on a relevant data set: x_1, x_2, …, x_n. These data values are random and are assumed to be mutually independent. The *parameter* to be estimated is a *fixed*, unknown *constant*.

One of the vital properties of a point estimator is that it should be **unbiased.** That is, it should be close to the population parameter it is trying to estimate, on average.

1.2 Sample Variance

We have seen in the previous lab that the **sample mean** $\bar{X}$ is an ideal candidate **estimator**, or **statistic**, for estimating the **population mean** μ (a parameter). What about the population variance σ^2 (another parameter)?

Ideally, we should have some results that say that the sample variance S^2 is close to σ^2 on average. Recall the formula for the sample variance:

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2$$

In this week's lab, we will use some simulations to check whether S^2 is really a good estimator of σ^2, how good it is, and how it compares to other candidate estimators. To do this, we will need a few more tools in our arsenal for R.

2.0 `For Loops` and Other Useful R Commands

2.1 `for loops`

A `for loop` is used to run a certain block of code a certain number of times.

```
for( "variable" in "vector" ) {
      do stuff
      }
```

For example:

```
for( i in 1:k ) {
      var[i] <- stuff
      }
```

First, the variable `i` is initialized to `1`. Then it goes through the block of code following the for statement. So, on the first pass through the code, we are saying `var[1] <- stuff`. Then, R will go back and assign `i` to be the second element of `1:k`, which is `2`. I.e., we increment the variable `i` by `1`. Then it will set `var[2] <- stuff`. This process repeats until `i = k`, in which it runs its final iteration. As you can imagine, for loops are very powerful – we can use the change value of `i` in creative ways. You can even imbed multiple for loops inside each other; however, this is usually fairly slow. For loops will often give the 'easiest' way to solve certain problems, but not necessarily the fastest or most efficient. For this class, we won't worry about efficiency too much.

2.2 Other useful R Commands

Type the following code into the console:

```
x <- matrix( 1:20, nrow = 5, ncol = 4 )
```

1. You can use `sum` command to find the sum of a data set. Calculate the sum of the entries in each of the four columns of `x` using a `for loop`. Recall that you can access the i^{th} column from the matrix `x` by writing `x[,i]`.
2. Calculate the sum of entries in each of the five rows of x using a `for loop`.
3. The functions `rowSums()` and `colSums()` can be used to calculate the sums of a matrix's rows and columns. Compare your answers in 1 and 2 with the output you get from these functions – they should be the same.
4. `'rnorm'` is the R command for generating random draws from a Normal distribution. Type `?rnorm` in your R console. What inputs does it take?

3.0 Pre-Lab Exercise

3.1 A Simulation

The Rockwell hardness of a certain metal is known to be Normally distributed with a mean of 105 and a standard deviation of 12. You are interested in estimating the *variance* in the Rockwell hardness measurements using a sample of the metal.

We can do a simulation and use this simulated data to calculate *point estimates* of the true population variance. Let's draw 100 samples of size 10 from the Normal distribution identified above, and calculate the *sample variance* for each sample.

First, let's give symbols to *sample size*, *number of samples*, and *parameters of the distribution*, so that we can change them easily to make other simulations under different conditions, keeping in mind that we can use a `for loop`, as well as that `rnorm` command takes the **standard deviation** (σ), not variance (σ^2), as the input. NA's in the below code mean that matrices and vectors will be created, but no data will be stored inside.

```
n <- 10
N <- 1000

mu <- 105
sigma <- 12

## Create an empty matrix to store the samples in and an empty vector
to store sample variances

samples <- matrix(NA, nrow = N, ncol = n)

sample.variances<-rep(NA, N)
```

We can use a for loop to generate the samples and store in the matrix as well as calculate the sample variance of each sample.

```
for (i in 1:N) {
samples[i,] <- rnorm(n, mean=mu, sd=sigma)
deviation <- samples[i,] - mean(samples[i,])
sample.variances[i] <- (1/(n-1)) * sum((deviation)^2)
}
```

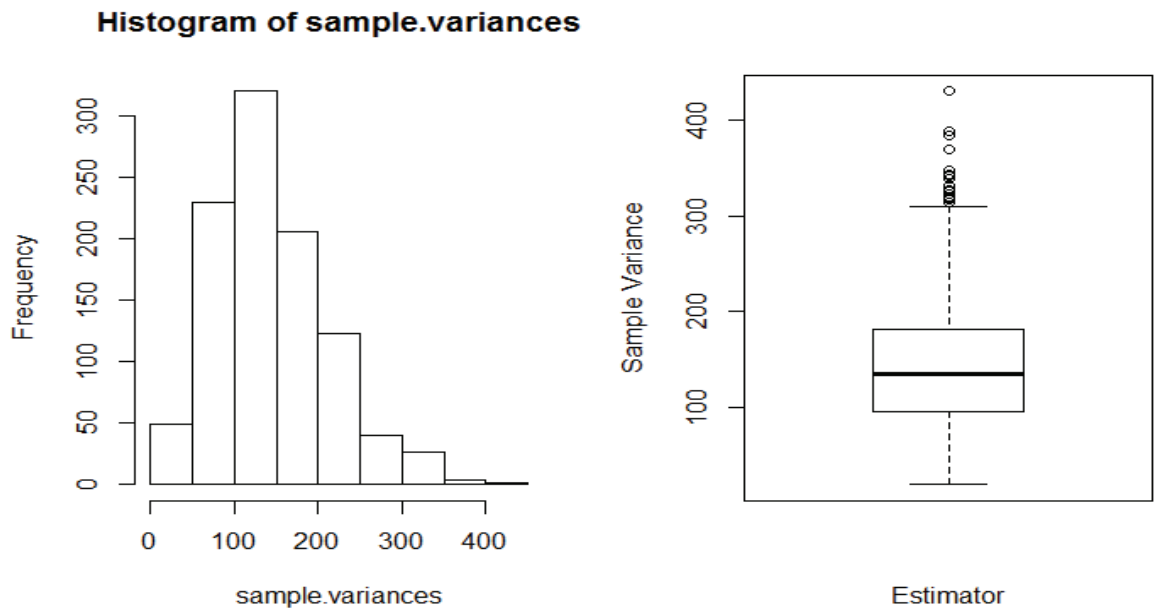
Is the ***mean*** *of the sample variances* (a statistic) the same as the *population variance* (the parameter) of the distribution that you sampled from? If not, by how much does it differ?

```
mean(sample.variances)
```

Sketch a histogram and a box plot of the sample variances. Which plot helps to visualize the shape of the distribution? Which plot helps to visualize the centre and spread of the distribution?

hist(sample.variances)

boxplot(sample.variances, `xlab = "Estimator", ylab = "Sample Variance"`)



3.2 Food for Thought

Answer the following questions. You can use the results you obtained in the simulation to answer most.

1. Is the shape of the histogram similar to the shape of any of the probability distribution functions you have come across so far, such as Uniform, Exponential or Normal? If *yes*, which one(s)? If *no*, how does this shape differ?
2. Would the shape of the histogram of sample variances change if you drew 1000 samples instead of 100?
3. Can you suggest other candidates for an estimator of *population variance*?
4. When you estimate a **parameter** (such as the *population variance* σ^2) using a **point estimate** (such as the sample variance S^2), you only obtain a single value. What disadvantages do you see in using a point estimate?
5. Do you recall how to draw *side-by-side box plots*?

**Think of at least one thing you would do to prepare better for the next lab exercise.