# Mathematical Statistics II

## Statistical Computing Activity: Module 3

One purpose of these Statistical Computing Activities (SCAs) is to give you a chance to explore statistics when the random variables do not follow a Normal distribution. Another purpose is to give you more skills in thinking about the randomness that is life.

Usually, like here, these SCAs will have a theme and several problems dealing with that theme or purpose. The reason for that extra layer of complexity is to tie what we do in the class with what we can use these techniques for in our lives as statisticians and/or consultants and/or full members of a democratic society.

⋆⋆⋆

**The Procedure.** Let $X$ and $Y$ be random variables, each having a Normal distribution. That is, let $X \sim \mathcal{N}(\mu_x;\ \sigma_x)$ and $Y \sim \mathcal{N}(\mu_y;\ \sigma_y)$. This is the usual assumption, allowing us to calculate exact confidence intervals. Let us not know either the means or the variances. Let us also take a sample of size $n_x$ and $n_y$ from each.

**Problem 1:**
Calculate the maximum likelihood estimators of $\mu_x$, $\mu_y$, $\sigma_x$, and $\sigma_y$.

**Problem 2:**
Note that the MLE for the variance is not an unbiased estimator. In fact, calculate the exact bias of the MLE estimator of $\sigma_x^2$.

**Problem 3:**
Calculate the distributions of $X - Y$ and $\overline{X} - \overline{Y}$.

**Problem 4:**
Since we want to estimate the difference in the population means, only one of those two distributions will be helpful. Using that distribution, determine the formula for confidence intervals on $\mu_x - \mu_y$.

**Problem 5:**

Check your work. Check that you actually created an appropriate confidence interval for $\mu_x - \mu_y$. To do this, let $n_x = 5$ and $n_y = 8$. Let $\mu_x = 0$, $\mu_y = 0$, $\sigma_x^2 = 1$, $\sigma_y^2 = 4$, and $\alpha = 0.05$. Perform the test using 10,000 iterations. Check that the coverage is close to $1 - \alpha$.

Here is some code that may help. This code generates a random vector of length 15 from a Normal(3,1) distribution and a random vector of length 10 from a Normal(3,11) distribution. It the calculates a 95% confidence interval for the difference in samples. After dong this 10,000 times, it determines how many of those confidence intervals cover the true value of 0.

```
nx=10; ny=18
mux=3; muy=3
s2x=1; s2y=11
alpha=0.05

lcb = numeric()
ucb = numeric()

for(i in 1:1e4) {
  x = rnorm(nx, m=mux, s=sqrt(s2x))
  y = rnorm(ny, m=muy, s=sqrt(s2y))

  pe = mean(x)-mean(y)
  se = sqrt( var(x)/nx + var(y)/ny )

  cv = -qt(alpha/2, df=nx+ny-2)

  lcb[i] = pe - cv*se
  ucb[i] = pe + cv*se
}

covered = (lcb*ucb)<0

sum(covered)
mean(covered)
```

The last two lines give the number of intervals covering 0 and the proportion of all intervals covering 0. This last number should be close to 95%.

You should be able to modify this script to meet your needs. By the way, Section 5.7 should be helpful in terms of the formulas.

**Problem 6:**

Estimate a 95% confidence interval for the coverage by using the Binomial procedure:

```
binom.test(x, n)
```

Here, set `x` equal to the number of times the parameter was covered and `n` equal to the number of iterations. If the resulting confidence interval includes $1 - \alpha$, then you do not have evidence that your procedure is wrong. Otherwise, you do.