# STA238 Tutorial 3

Luis Ledesma

2023-02-08

## 1 Announcements

- You can upload your work on Crowdmark from the end of the tutorial session to 5pm Friday of that week.
- All questions must be solved using RStudio.

## 2 Recall: Last tutorial

Last tutorial: we proposed different estimators of a parameter and we determined which ones were unbiased. Out of the unbiased estimators, we also identified the one with the lowest variance.

Main takeaways:

1. An estimator $\hat{\theta}$ of $\theta$ will be a random variable, not a fixed parameter. In contrast, $\theta$ will be a fixed parameter (non-random).
2. An unbiased estimator $\hat{\theta}$ of $\theta$ will have the characteristic that $E(\hat{\theta}) = \theta$.
3. A sample has the characteristic that the random variables are independent and identically distributed. If $X \perp Y$ (independent), then $E(X + Y) = E(X) + E(Y)$ and $\text{var}(X + Y) = \text{var}(X) + \text{var}(Y)$.

## 3 Tutorial activity

In broad strokes, we want to:

1. Given values from a sample, we want to compute some point estimates from the sample.
2. Assuming that the sample follows a particular distribution, computing some probability ranges.
3. Conducting a simulation study to verify the property that the sum of normal random variables will also be a normal random variable.

### 3.1 Computing point estimates from a sample

Given the sample in the problem, we have:

```
samp <- c(.83, .88, .88, 1.04, 1.09,
          1.12, 1.29, 1.31, 1.48, 1.49, 1.59, 1.62,
          1.65, 1.71, 1.76, 1.83)
```

If we want point estimates of the mean $\mu$ and variance $\sigma^2$, we could compute $\hat{\mu}$ and $s^2$. In R, the commands for these are given by `mean` and `var` (`sd` will just be the square root of `var`).

```r
sampmean <- mean(samp)
sampvar <- var(samp)
sampmean
```

```
## [1] 1.348125
```

```r
sqrt(sampvar/length(samp))
```

```
## [1] 0.08463263
```

Recall that the estimate for the standard error will be given by $\frac{s}{\sqrt{n}}$.

## 3.2 Computing probability estimates from the given sample

We can assume that the sample of coating thickness will be approximately normal. This allows us to assume that it will follow a distribution given by $X \sim N(\hat{\mu}, s)$ (where the parameters of the normal are the ones obtained from our point estimates).

If we want to calculate a point estimate which separates the largest 10% from the remaining 90%, we may be interested in computing the value $x'$ such that $P(X < x') = 0.9$.

```r
qnorm(0.9,mean=sampmean,sd=sqrt(sampvar))
```

```
## [1] 1.781969
```

Alternatively, by properties of the normal distribution, we could use:

```r
sampmean+qnorm(0.9)*sqrt(sampvar)
```

```
## [1] 1.781969
```

If we want to estimate $P(X < 1.5)$, then we would use the function `pnorm`.

```r
pnorm(1.5,mean=sampmean,sd=sqrt(sampvar))
```

```
## [1] 0.6731508
```

Note that in this case we are computing the lower tail of the distribution function.

## 3.3 Sums of independent normal random variables

Let $X_1, \ldots, X_30$ be independent and follow a normal random distribution with mean 1 and variance 4. Suppose we simulate 40000 values from $X_1 + \cdots + X_30$.

```r
B <- 40000
n <- 30
simvalues <- matrix(0,nrow=n,ncol=B)

for(i in 1:B){
  simvalues[,i] <- rnorm(n,mean=1,sd=2)
}
```

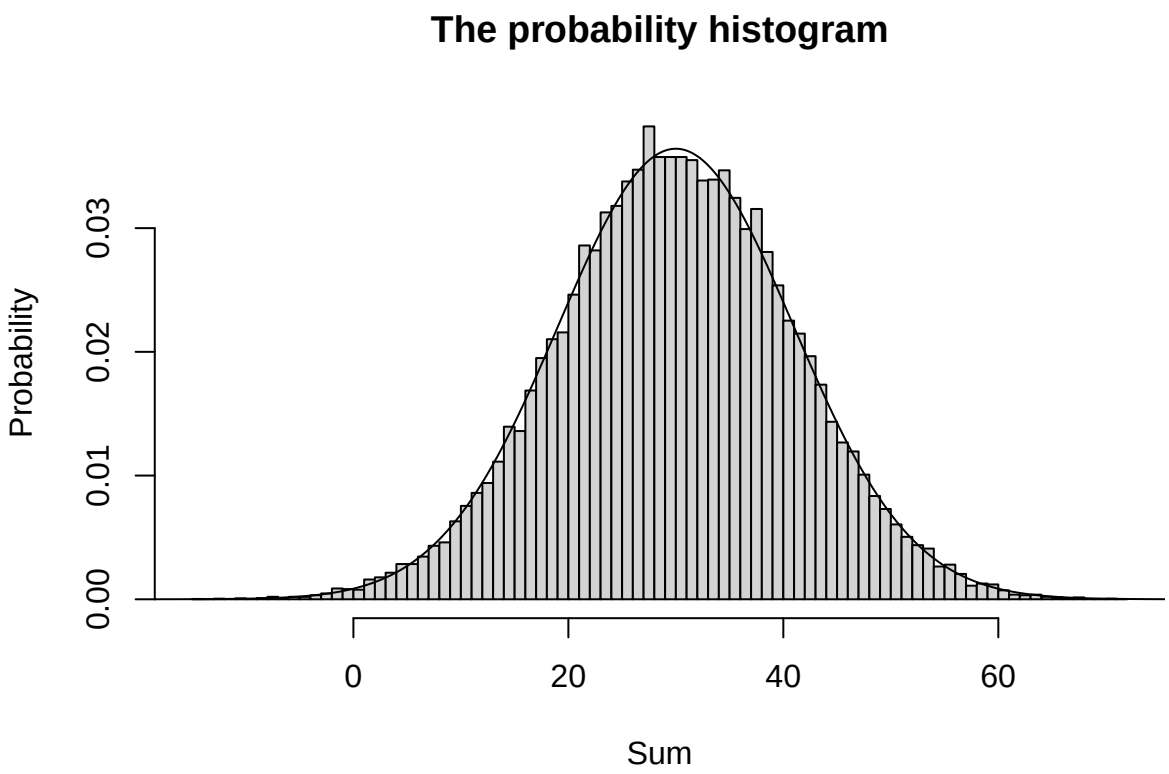
We should expect that $X_1 + \cdots + X_30$ will follow a normal distribution with mean 30 and variance 120, i.e. $N(\mu_1 + \cdots + \mu_{30}, \sigma_1^2 + \cdots + \sigma_{30}^2) = N(30, 120)$, **as the $X_i$ are independent**.

```r
simsums <- colSums(simvalues)
```

If we plot a histogram of `simsums` and compare it to a plot of the density function of $N(30, 120)$, we should expect to get something similar.

```r
xvals <- seq(30-50, 30+50, by = .1)
yvals <- dnorm(xvals, mean = 30, sd = sqrt(120))

hist(simsums,prob=TRUE,main="The probability histogram",xlab="Sum",ylab ="Probability",breaks=100)
lines(xvals,yvals)
```




The argument `prob=TRUE` will make it so that we get the probability in the y-axis rather than the frequency in the histogram.