Lecture 20: More Simulation

Statistical Computing, 36-350

Wednesday December 2, 2015

Warm up: simple examples

t random variables, with 5 df:

```
n = 1000
t.draws = rt(n,df=5) # t5 random variables
mean(t.draws)   # Check: mean approx 0
```

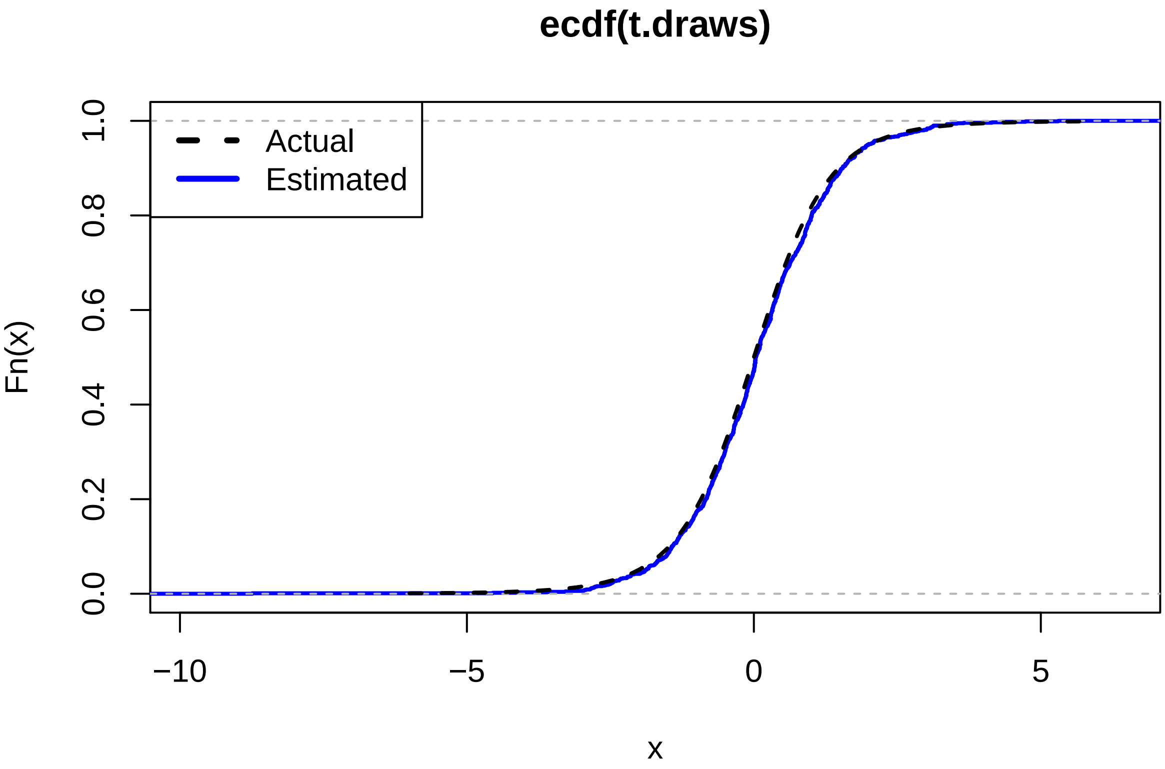
```
## [1] 0.07083142
```

```
var(t.draws)    # Check: variance approx 5/3
```

```
## [1] 1.55954
```

Empirical distribution function:

```
# Let's plot the empirical distribution function
t.ecdf = ecdf(t.draws)
plot(t.ecdf, lwd=2, col="blue")
# Let's also plot the actual distribution function on top
x = seq(-6,6,length=100)
lines(x, pt(x,df=5), lwd=2, lty=2)
legend("topleft", lty=c(2,1), lwd=3,
       col=c("black","blue"), legend=c("Actual","Estimated"))
```



```
# Note: t.ecdf is an actual function! We can evaluate it
t.ecdf(1)
```

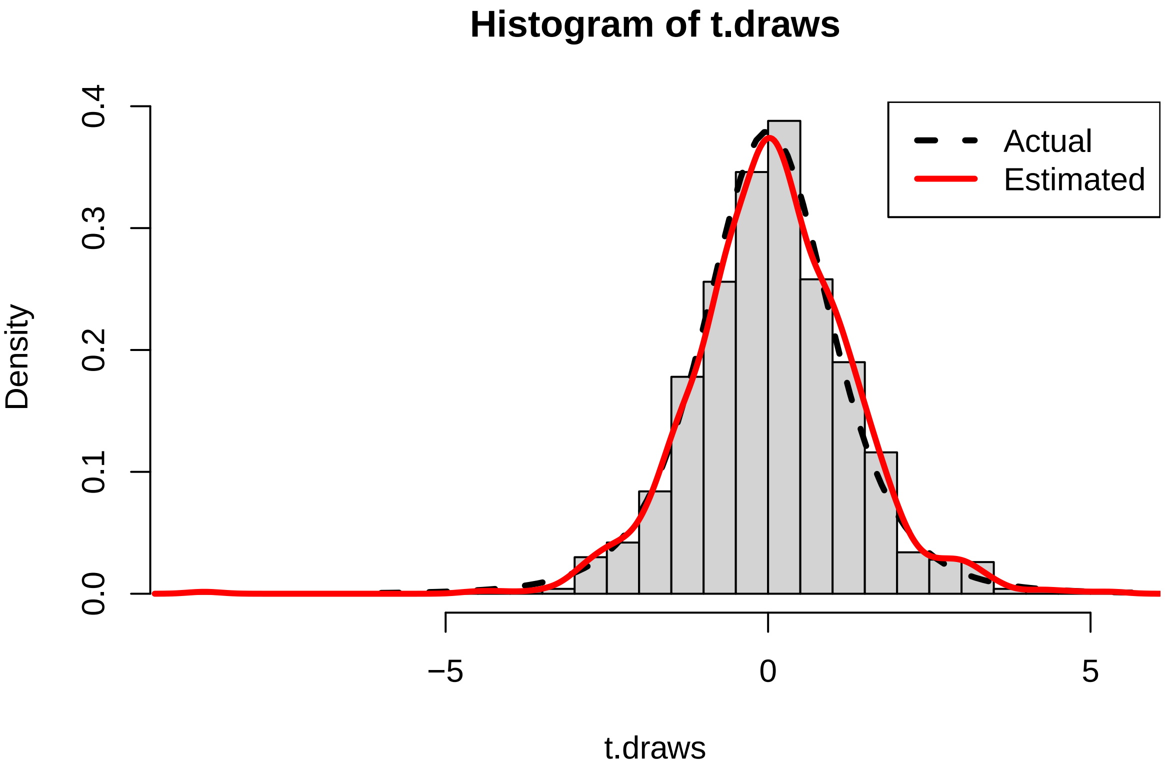
```
## [1] 0.797
```

```
pt(1,df=5)
```

```
## [1] 0.8183913
```

Histogram and empirical density functions:

```
# Let's plot a histogram of our t5 random variables
hist(t.draws, breaks=40, probability=TRUE, col="lightgray")
# Let's also draw the t5 density on top
x = seq(-6,6,length=100)
lines(x, dt(x,df=5), lwd=3, lty=2)
# Let's also draw an empirical (estimated) density on top
lines(density(t.draws), col="red", lwd=3)
legend("topright", lty=c(2,1), lwd=3,
       col=c("black","red"), legend=c("Actual","Estimated"))
```



Advanced fact for you to remember (just if you are curious): it's a lot easier to estimate a distribution function, than to estimate a density function!

Why simulate?

R gives us unique access to great simulation tools (unique compared to other languages). Why simulate? Welcome to the 21st century! Two reasons:

- Often times simulations can be easier than hand calculations
- Often times simulations can be made more realistic than hand calculations

Drug effect sizes

- Suppose we had a model for the way a drug effected certain patients
- E.g., all patients will undergo chemotherapy. We believe those who aren't given the drug experience a reduction in tumor size of percentage

$$X_{\text{no drug}} \sim 100 \cdot \text{Exp}(\text{mean} = R), \quad R \sim \text{Unif}(0,1)$$

- And those who were given the drug experience a reduction in tumor size of percentage

$$X_{\text{drug}} \sim 100 \cdot \text{Exp}(\text{mean} = 2)$$

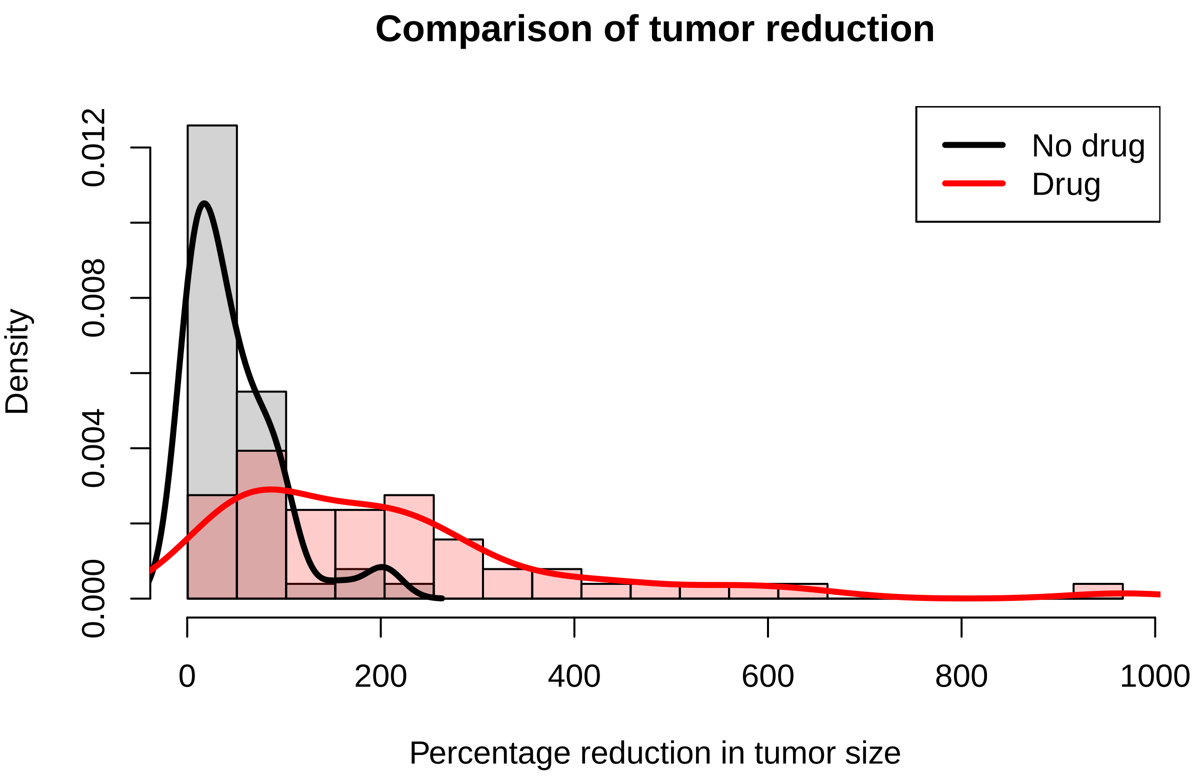
```r
# Simulate from model, supposing 50 subjects in each group
set.seed(0)
n = 50
mu.drug = 2
mu.nodrug = runif(n,min=0,max=1)
x.drug = 100*rexp(n,rate=1/mu.drug)
x.nodrug = 100*rexp(n,rate=1/mu.nodrug)
summary(x.drug)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.454  72.050 165.000 207.400 258.600 966.600
```

```r
summary(x.nodrug)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   0.4944  10.1800  33.7700  46.8000  73.2100 205.2000
```

```r
# Find the range of all the measurements together
x.range = range(c(x.nodrug,x.drug))
# Here we are manually defining breaks for the hist() function
breaks = seq(min(x.range),max(x.range),length=20)
# Produce a histogram of the non drug measurements
hist(x.nodrug, breaks=breaks, probability=TRUE, xlim=x.range,
     xlab="Percentage reduction in tumor size",
     col="lightgray", main="Comparison of tumor reduction")
# Plot a histogram of the drug measurements, on top
hist(x.drug, breaks=breaks, probability=TRUE, add=TRUE,
     col=rgb(1,0,0,0.2)) # Transparent color! r, g, b, alpha
lines(density(x.nodrug),lwd=3)
lines(density(x.drug),lwd=3,col="red")
legend("topright", lty=1, lwd=3, col=c("black","red"),
       legend=c("No drug","Drug"))
```



Consider the following:

- You work for a drug company that wants to put this new drug out on the market
- But in order to get FDA approval, your company must demonstrate that the patients who had the drug had **on average** a reduction in tumor size **at least 100 percent greater than** those who didn't receive the drug, or in symbols:
$$\overline{X}_{\text{drug}} - \overline{X}_{\text{no drug}} \geq 100$$
- Your drug company wants to spend as little money as possible. They want the smallest number n such that, if they were to run a clinical trial with n patients in each of the drug / no drug groups, they would likely succeed in demonstrating that the effect size (as above) is at least 100
- Of course, the result of a clinical trial is random; your drug company is willing to take "likely" to mean **successful with probability 0.95** (i.e., successful in 95 of 100 hypothetical clinical trials, though only 1 will be run in reality)

A thought experiment, first. Given the model for tumor reduction with and without drugs, is it even possible to see an average difference of 100 percent between the two groups?

The larger n gets, the closer the averages get to the (population) means under each model. Well

$$E[X_{\text{drug}}] = 100 \cdot E[\text{Exp}(\text{mean} = 2)] = 200$$

And

$$E[X_{\text{nodrug}}] = 100 \cdot E[\text{Exp}(\text{mean} = R)] < 100 \cdot E[\text{Exp}(\text{mean} = 1)] = 100$$

since, recall, $R \sim \text{Unif}(0, 1)$. So the difference in means is at least 100. This means that it's certainly possible for the difference in averages to be at least 100

Practice problems

Enter your unique ID here:

Work through the following problems (go ahead and fill in the code below)

```
# 1. Write a function around the simulation code above, that produces measurements in
# the drug and no drug groups. Your function should be called sim.drug.effect(), and
# it should take two arguments:
# - n, the number of subjects in each group, with a default value of 50
# - mu.drug, the mean for the exponential distribution that defines the drug
#   tumor reduction measurements, with a default value of 2
# Your function should return the average difference in tumor reduction between the
# subjects who received the drug, and those who didn't.
# Note: this function should NOT call set.seed(), you want its results to be random
```

```
# 2. For each value of n in between 5 and 200, perform the following. Run
# sim.drug.effect() a total of 100 times, with the given value of n; record
# the average difference in tumor reduction; and then count the number of
# successes, i.e., the number of times (out of 100) that this difference
# exceeds 100. Make a line plot which shows the number of successes as a
# function of n. Label the axes appropriately. What is the smallest n for
# which the number of successes exceeds 95?
# (Hint: for this task, a double for() loop would work just fine ...)
```

```
# 3. Now suppose your drug company told you they only had enough money to
# enlist n=20 subjects in each of the drug / no drug groups, in their clinical
# trial. They then asked you the following question: how large would mu.drug
# have to be, the mean proportion of tumor reduction in the drug group, in order
# to have probability 0.95 of a successful drug trial? Run a simulation,
# much like your simulation in problem 2, to answer this question.
# (Hint: now you will want to let mu.drug vary over a wide range, say 1.5 to 5,
# and for each value, simulate data with n=20 from the drug / no drug groups,
# compute difference of average tumor reduction percentages, etc.)
```

```
# 4. So, it turns out that the drug company can actually control mu.drug, the
# mean proportion of tumor reduction among the drug subject, by adjusting the
# dose concentration of some secret special chemical. But there is no free
# lunch: the higher concentration of this secret chemical, the more likely
# a subject is to have liver failure. In particular, suppose that:
# - people on the drug die with probability sqrt(mu/4000)
# - the FDA has a policy that if 2 subjects die in a clinical trial, then
#   the trial is shut down
# - in this case, the trial is clearly not counted as a success (even if
#   the average difference in tumor reduction percentage was huge, between
#   surviving members of the two groups)
# As in problem 3, suppose that the drug company only has enough money to enlist
# n=30 people in each of the drug / no drug groups in their clinical trial. Adapt
# your simulation from problem 3 to incorporate the fact that patients can die
# from liver failure, in the drug group. Count the number of successes (out of
# 100) as a function of mu.drug. Is there any hope here? I.e., is there a value
# for which we have at least 95 successes?
```

```
# Bonus. Suppose that the model for tumor reduction for non drug subjects is pretty
# accurate, but that for the drug subjects can be refined. Suppose in particular
# that there are three types of patients:
# - non-responders, making up 25% of the general population, for whom mu.drug is
#   distributed as Unif(0,1)
# - regular-responders, making up 65% of the population, for whom mu.drug = 2
# - super-responders, making up 10% of the population, for whom mu.drug = 4
# Update your function sim.drug.effect() from problem 1 to accommodate this more
# accurate model the drug group of patients. Rerun your simulation from problem 2
# and answer the same questions as before. Does the required sample size n get
# smaller or bigger?
# (Hint: for simulating from this more accurate model, you'll want to flip a coin
# each time to determine what type of patient you are looking at: a non-responder,
# regular-responder, or super-response. The coin should be three-sided and have
# probabilities 0.24, 0.85, and 0.1 of landing on each side; you can use runif()
# to build such a coin ...)
```