8. Simulation by replicating a calculation

Make random number generation repeatable

`set.seed(seed)`, for integer `seed`, sets starting point of (pseudo-)random number generation. e.g.

```
a = rnorm(1); b = rnorm(1); a == b
set.seed(0); a = rnorm(1); set.seed(0); b = rnorm(1); a == b
```

Repeat a calculation n times

`replicate(n, expr)` returns a vector (or matrix or array) of `n` evaluations of `expr`. e.g.

```
x = replicate(n=4, expr=rnorm(1))       # 4 random samples of size 1
y = replicate(n=4, expr=rnorm(3))       # 4 random samples of size 3
z = replicate(n=4, expr=mean(rnorm(3))) # 4 means of samples of size 3
```

`expr` can be compound in curly braces, `{ ... }`; its value is that of its last expression. e.g.

```
w = replicate(n=4, expr={ mu=7; sigma=3; x=rnorm(n=3, mean=mu, sd=sigma); mean(x) })
```

Distributions

Check `?distributions`. (Recall prefixes `d, p, q, r` for *density*, *probability*, *quantile*, *random*.)

Let's simulate a few distributions.

- $N(\mu, \sigma)$: First, confirm that $\bar{x}$ is close to μ and that s is close to σ:

  ```
  mu = 7; sigma = 3; mean(x <- rnorm(n=1000, mean=mu, sd=sigma)); sd(x)
  ```

  Second, the *Central Limit Theorem* (CLT) says that for a large sample from (almost) any distribution with finite μ and σ, $\bar{X} \approx N(\mu, \frac{\sigma}{\sqrt{n}})$.

  e.g. Consider $U(0,1)$, which has $\mu = \frac{\max - \min}{2} = \frac{1}{2}$ and $\sigma = \sqrt{\frac{(\max-\min)^2}{12}} = \sqrt{\frac{1}{12}}$. Simulate CLT by finding many sample means from samples from $U(0,1)$:

  ```
  curve(dunif(x, min=0, max=1), from=-0.1, to=1.1, ylim=c(0,8), lty=2) # U(0,1)
  n =  30 # sample size (also try n=1 to see CLT fail)
  N = 100 # number of samples
  x.bars = replicate(n=N, expr=mean(runif(n=n, min=0, max=1))) # vector of sample means
  mean(x.bars) # should be near 1/2
  sd(x.bars)   # should be near sqrt(1/12)/sqrt(n), about .0527
  curve(dnorm(x, mean=1/2, sd=sqrt(1/12)/sqrt(n)), from=0, to=1, lty=3, add=TRUE) # CLT
  lines(density(x.bars), lty=1) # sampling distribution of bar(x)
  rug(x.bars)
  legend(x="topright", legend=c("U(0,1)", "CLT", expression(bar(X))), lty=c(2,3,1))
  ```

- t_{n-1}: For a random sample $X_1, \ldots, X_n$ from $N(\mu, \sigma)$, the quantity $T = \frac{\bar{X}-\mu}{s/\sqrt{n}}$ follows the *Student's t* distribution with $n-1$ degrees of freedom, denoted t_{n-1}.

  e.g. Simulate t_{n-1} for $n = 6$:

```
n =   6 # sample size
N = 100 # number of samples
mu = 7
sigma = 3
t = replicate(N, { x=rnorm(n, mean=mu, sd=sigma); (mean(x) - mu)/(sd(x)/sqrt(n)) })
plot(density(t)) # sampling distribution of T ~ t_{n-1}
rug(t)
curve(dt(x, df=n-1), lty="dashed", add=TRUE) # true t_{n-1}
curve(dnorm(x, mean=0, sd=1), lty="dotted", add=TRUE) # add N(0, 1) for reference
legend(x="topright", legend=c(expression("true "*t[n-1]), "simulated t", "N(0, 1)"),
       lty=c("dashed", "solid", "dotted"))
```

- χ^2_n: If $Z_1, \cdots, Z_n$ are independent, $N(0,1)$ random variables, then $X^2 = \sum_{i=1}^n Z_i^2 \sim \chi^2_n$. ...
- F_{n_1,n_2}: If $X \sim \chi^2_{n_1}$ and $Y \sim \chi^2_{n_2}$ are independent, then $\dfrac{X/n_1}{Y/n_2} \sim F(n_1, n_2)$. ...

What is a P-value?

A P-value is the probability, assuming H_0 is true, of getting data, as summarized by the test statistic, more extreme than the sample data. e.g. Guinness says pouring a glass should take 119.5 seconds. Here's a random sample of times from a server:

```
x = c(118, 121, 113, 116, 117, 112, 113)
```

Is this server pouring correctly? Test $H_0 : \mu = 119.5$ vs. $H_1 : \mu \neq 119.5$: `(out = t.test(x, mu=119.5))`

Simulate P-value by seeing how often `t`, from random samples, is greater than `out$statistic`:

```
mu = 119.5
sigma = sd(x)
n =  length(x) # sample size
N = 1000 # number of replicates
t = replicate(N, { x=rnorm(n, mean=mu, sd=sigma); (mean(x) - mu)/(sd(x)/sqrt(n)) })
more.extreme = (abs(t) > abs(out$statistic))
(simulated.p.value = sum(more.extreme) / N)
out$p.value

plot(density(t), main=bquote(.(N) * " Simulated t statistics")) # visualize P-value
rug(t)
points(x=out$statistic, y=0, pch=19, col="red")
text(x=out$statistic, y=.02, labels="out$statistic")
```