Let the parent distribution be described by a p.d.f. $f(x)$ with c.d.f. $F(x)$. Let X be a random variable with the parent distribution. We have $\mathsf{E}(X) = \mu$ and $\mathsf{SD}(X) = \sigma$. A random sample from the parent distribution, $X_1, X_2, \ldots, X_n$ is an iid collection of random variables each distributed like X. A statistic, T, is a function of the random sample. It too is a random variable and so has its own distribution called the sampling distribution. In addition to the distribution of T, we may be interested in it's mean and standard deviation.

1 Finding random samples

R has built in functions to plot a p.d.f and to find a sample from a parent distribution. The "d-functions" will give the density, and the "r-functions" will help find a sample. For example, if the parent distribution is normal with mean 5 and standard deviation=2 the following will plot the density, take a sample, plot it as lines and make a density estimate from the sample.

```
> curve(dnorm(x,mean=5,sd=2),-1,11)
> x = rnorm(25,mean=5, sd = 2)
> rug(x)
> lines(density(x), lty=2)
```

R has other distributions, but they are used similarly. The stem for the normal is `norm`. The functions are "r" or "d" connected to the stem. For example, these are the "d-functions" for some families of distributions: `dbinom(x,size=n,prob=p)` for the binomial(n,p), `dexp(x,rate=lambda)` for exponential with mean $1/\lambda$, `dt(x,df=df)` for the t-distribution with `df` degrees of freedom (later) and `dlnorm(x,meanlog=mu,sdlog=sigma)()` for the log-normal distribution.

Many of the parameters have a default. For example, `dnorm` uses mean 0 and standard deviation 1.

Exercise 0.1 For each of these stems: `exp`, `unif` and `t` (with `df=3`) plot the density, take a sample of size 25 and plot it with `rug()` and plot a density estimate.

Exercise 0.2 For `norm` plot the density, then take samples of size 25,100 and 1000 and plot density estimates. Does the estimate get better as the sample size increases?

2 sampling distributions

When we have a statistic, we can sometimes work out a sampling distribution. Even when we can it may not lend itself to ready analysis. For example, we find this formula for the Median when X_i are from a continuous distribution

$$f_M(x) = \binom{2k+1}{1}\binom{2k}{k} F(x)^k f(x)(1-F(x))^k$$

We could plot this using the "p-functions", but lets look at a *simulation*. That is, we will generate our own random samples from this distribution. There won't be a built in function, so we'll need to so some extra work.

For example, this will simulate the median when $n = 15 = 2(7)+1$, $m = 250$ times for X_i normal with mean 0 and standard deviation 1.

```
> res = c()                        # a place to store the data
> k = 7; n = 2*k+1; m = 250        # parameters
> for(i in 1:m) res[i] = median(rnorm(n)) # for loop
> hist(res)                        # view results with histogram, qqplot
> qqnorm(res)
```

The key line is the `for` loop which loops over 1 through `m` taking a new sample each time and storing its median into `res[i]`.

If you run this, the graphs will indicated the distribution is approximately normally distributed.

Exercise 0.3 Repeat the above with `rexp(n)` instead of `rnorm(n)`. Is there a difference in the sampling distribution from before? What about if $k = 100$ is the sampling distribution approximately normal?

Exercise 0.4 To simulate the sample standard deviation, S, for normal data, you would replace `res[i] = median(rnorm(n))` with the similar `res[i] = sd(rnorm(n))`. Do so with $n = 15$ and describe the shape of the sampling distribution of S.

3 easy editing

Typing these commands again and again can be a chore. A trick to help out is to store them inside a function which can be easily edited. Here's how.

First make a function, `simit`, using the `function()` keyword as follows

```
> simit = function ()
```

Then edit the function using the `edit()` command. Notice, we need to reassign the result of edit to the function

```
> simit = edit(simit)
```

This opens up a text editor, make changes to leave the file like this

```
simit = function()  res = c() k = 7; n = 2*k+1; m = 250 for(i in 1:m) res[i] = median(rnorm
```

Save the function (`R` has named it already with a secret name) and exit the editor. Running the function as follows will make the simulation and store them in `res` and plot the histogram. You can then plot the quantile plot as

```
> simit()                          # run the function
> qqnorm(res)                      # plot res
```

Repeat by changing the function definition.

4 The Central Limit Theorem (CLT)

Perhaps the most important statistic is $\bar{X} = (X_1 + \cdots X_n)/n$. It has a distribution that can be explained using complicated expressions involving f and F, however, it also has a limit that can be easily explained.

First, the mean and standard deviation of $\bar{X}$ are easy to find. The mean is always easy for a sum, the standard deviation is as X_i are assumed to independent. These values are

$$\mathsf{E}(\bar{X}) = \mu_{\bar{X}} = \mu, \quad \mathsf{SD}(\bar{X}) = S_{\bar{X}} = \sigma/\sqrt{n}.$$

That is the mean of the sample average is also the population mean, but the standard deviation of the sample average is the population standard deviation *divided* by $\sqrt{n}$.

To illustrate, this will plot simulations of the sample average for normal data for $n = 4, 16, 64$ and 256

```
> n = c(4,16,64,256)
> res = c()
> plot(0,0,pch=" ",xlim=c(-2,2),ylim=c(0,12))
> for (i in n) + for(j in 1:m) res[j] = mean(rnorm(i))+ lines(density(res))+
```

If you type this in, notice how the center is always at $\mu = 0$, but the spread gets cut 1/2 each time.

The graph of the density estimate looks bell-shaped which is due to the fact that when $X_1, X_2, \ldots, X_n$ are iid *normal* then the sampling distribution of $\bar{X}$ is also normal with the mean and standard deviation above.

The amazing fact is that no matter what the parent distribution is, as long as it has a μ and σ the distribution of $\bar{X}$ is *approximately normal* if n is large enough. This is the central limit theorem which states that

$$\lim_n P(a \leq \frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}} \leq b) = P(a \leq Z \leq b),$$

where Z is a standard normal.

Exercise 0.5 Simulate $\bar{X}$ when $n = 25$ and X_i are uniform on $[0,1]$ (the default for `runif`. Is $\bar{X}_{25}$ approximately normal? That is, is n large enough in this case?

Exercise 0.6 Simulate $\bar{X}$ when $n = 25$ and X_i are exponential with parameter $\lambda = 1/10$. (Use `rexp(n,rate=1/10)` Is $\bar{X}_{25}$ approximately normal? That is, is n large enough in this case? In not, try with $n = 100$. Is it large enough now?

4.1 Monte Carlo

The central limit theorem can be used to "integrate" functions. Suppose, you need to integrate

$$\int_0^{\sqrt{\pi}} \sin(x^2) dx$$

No anti-derivative is available, so you need to use a numeric method. The Monte Carlo method will also work. Basically, it says you can integrate this by looking at the sampling distribution of a random variable.

How? to integrate $\int_a^b g(x)dx$ let X be uniform on $[a,b]$ and look at the random variable $(b-a)g(X)$. It has expectation given by

$$\mathsf{E}((b-a)g(X)) = (b-a)\int g(x)P(X=dx) = (b-a)\int_a^b g(x)\frac{1}{b-a}dx = \int_a^b g(x)dx.$$

That is the mean of $(b-a)g(X)$ is the value we want. By taking a sample of $(b-a)g(X_i)$ and finding its sample mean, $(b-a)\overline{g}$, we have a random variable with mean that gives the number we want and standard deviation bounded by $(b-a)max(g^2)/\sqrt{n}$. The value of n can be chosen so large that the estimate $(b-a)\overline{g}$ is basically the integral.

Exercise 0.7 Try this out with $n = 100$ and $n = 1000$. What is your guess for the integral

$$\int_0^{\sqrt{\pi}} \sin(x^2)dx?$$

Check yourself with the R command `integrate()`

```
> integrate(function(x) sin(x^2),0,sqrt(pi))
```