Functions in R

- R has many in-built functions.
- Writing new functions is also possible.
- It can be constructed using `function`

Functions in R: Syntax

- say we are trying to find the `mean` of vector

```
> ourmean = function(x) {
+ sum(x)/length(x)
+ }
```

- the `function` will return the last computed value unless the body calls for a specific return value.

```
> x = c(1,2,3,4,4,5,5,5,5)
> ourmean(x)
[1] 3.777778
```

Functions in R

- Try to use in-built functions -R
- It does take effort to write a useful funciton using `function` that provides one single number.

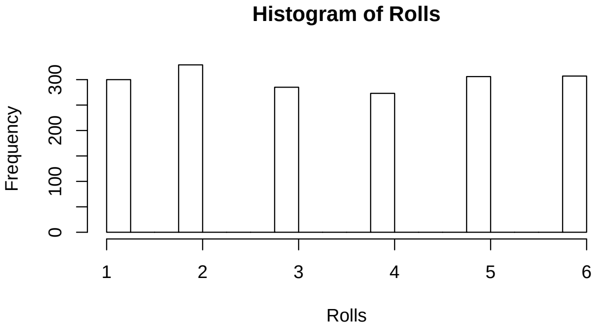
Sampling from a given distribution

- we can use the `sample` function.
- takes a sample of the specified size (specified by `size`) from the elements of x using either with or without replacement (specified by `replace`).
- The optional `prob` argument can be used to give a vector of weights for obtaining the elements of the vector being sampled.

```
> x = c(1,2,3,4,5,6)
> probx= c(1/6,1/6,1/6,1/6,1/6,1/6)
> Rolls=sample(x, size=1800,  replace=T, prob=probx)
```

Uniform(1,2,3,4,5,6)

```
> table(Rolls)
Rolls
  1   2   3   4   5   6
300 329 285 273 306 307
> hist(Rolls,breaks = seq(1,6, by=0.25))
```



Functions in R: Variance of Uniform

- Let us try to compute the `variance` of x

```
> x
[1] 1 2 3 4 5 6
> ourvariance = function(x) {
+  sum((x -ourmean(x))^2)/length(x)
+ }
```

- Note that this differs from sample variance in the normalisation.

Uniform(1,2,3,4,5,6)

```
> var(Rolls)
[1] 2.980381
> ourvariance(x)
[1] 2.916667
```

- `ourvariance` gives the variance of the uniform random variable.

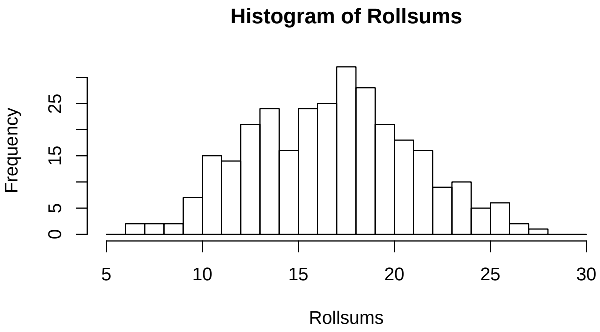
Sums of Rolls

Suppose we wish to simulate in R the experiment that we did in class of Rolling a die and noting down its sum. We can use the `sample` , `matrix` and `apply`.

```
> x = c(1,2,3,4,5,6)
> probx= c(1/6,1/6,1/6,1/6,1/6,1/6)
> Rolls=sample(x, size=1500,  replace=T, prob=probx)
> Rollm=matrix(Rolls, 5)
> # above creates a matrix 5 columns and 30 Rows
> Rollsums = apply(Rollm, 2, sum)
```

Sums of Rolls

```
> table(Rollsums)

Rollsums
 7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
 2  2  2  7 15 14 21 24 16 24 25 32 28 21 18 16  9 10  5  6  2  1

> hist(Rollsums,breaks = seq(5,30, by=1))
```



This was the histogram that we got when we did the experiment of rolling a die 5times and noting down its sum.

Sampling distribution

Suppose we want to verify the below result via simulations:

Let $X_1, X_2, \ldots, X_n$ be an i.i.d. sample of random variables whose distribution has finite expected value μ and finite variance σ^2. Let $\bar{X}$ represent the sample mean. Then

$$E[\bar{X}] = \mu \quad \text{and} \quad SD[\bar{X}] = \frac{\sigma}{\sqrt{n}}.$$

Sampling distribution

```
> x = c(1,2,3,4,5,6)
> probx= c(1/6,1/6,1/6,1/6,1/6,1/6)
```

Let us generate 3 sets of data:

500,5000,150000 samples from x and probx.

```
> Rolls=sample(x,size=500,replace=T,prob=probx)
> Rolls5000=sample(x,size=5000,replace=T,prob=probx)
> Rolls150000=sample(x,size=150000,replace=T,prob=probx)
```

Sampling distribution

We split them up into sets of $5, 50, 5000$ rolls.

```
> Rollm=matrix(Rolls, 5)
> Rollm5000=matrix(Rolls5000, 50)
> Rollm150000=matrix(Rolls150000, 5000)
```

Thus each gives us sets of 100,100,30 trials respectively fo $5, 50, 5000$

Sampling distribution

Let us compute the the mean of each row which are of size $5, 50, 5000$

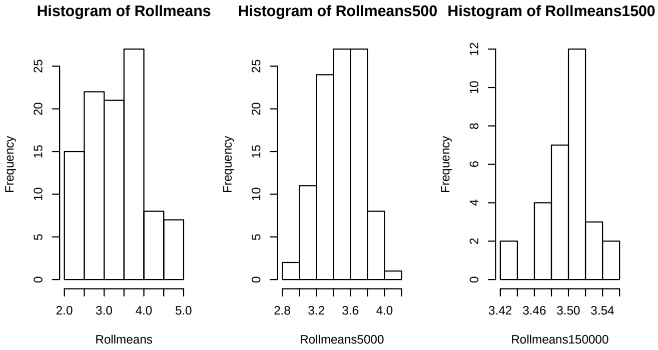
```
> Rollmeans = apply(Rollm, 2, mean)
> Rollmeans5000 = apply(Rollm5000, 2, mean)
> Rollmeans150000 = apply(Rollm150000, 2, mean)
```

Mean of Rolls

```
> table(Rollmeans)

Rollmeans
  2 2.2 2.4 2.6 2.8   3 3.2 3.4 3.6 3.8   4 4.2 4.4 4.8   5
  3   3   9  10   7   5  10  11  11   8   8   5   3   5   2

> table(Rollmeans5000)

Rollmeans5000
2.88    3 3.02 3.04 3.06 3.08  3.1 3.12 3.14 3.16 3.18  3.2 3.22 3.24 3.26 3.28
   1    1    1    1    1    1    1    1    1    1    1    2    3    2    1    2
3.32 3.34 3.36 3.38  3.4 3.42 3.44 3.46 3.48  3.5 3.52 3.54 3.56 3.58  3.6 3.62
   3    1    6    3    3    2    3    5    2    2    2    4    3    2    2    5
3.64 3.66 3.68  3.7 3.72 3.74 3.76 3.78  3.8 3.82 3.84 3.86 3.88  3.9 3.92 3.96
   1    6    4    4    2    1    2    1    1    2    1    1    1    1    1    1
4.02
   1

> table(Rollmeans150000)

Rollmeans150000
3.4326 3.4334 3.4644  3.474  3.477 3.4774 3.4826 3.4828 3.4844 3.4868 3.4902
     1      1      1      1      1      1      1      1      1      1      1
3.4912 3.4938 3.5006 3.5016 3.5028  3.503  3.505  3.507  3.508 3.5084 3.5102
     1      1      1      1      1      1      1      1      1      1      2
 3.511 3.5172 3.5268 3.5316 3.5334 3.5482  3.551
     1      1      1      1      1      1      1
```

Centered around 3.5

```
> par(mfrow=c(1,3))
> hist(Rollmeans)
> hist(Rollmeans5000)
> hist(Rollmeans150000)
```



Variance Reduction

Observe that there is real variance reduction in the sample means.

```
> ourvariance(x) # Variance of Uniform (1,2,3,4,5,6)

[1] 2.916667

> var(Rollmeans) # S^2, 100 Trials, mean of 5 Rolls

[1] 0.5527919

> var(Rollmeans5000)# S^2,100 Trials, mean of 50 Rolls

[1] 0.056544

> var(Rollmeans150000)# S^2, 100 Trials, mean of 5000 Rolls

[1] 0.0007484258
```

Central Limit Theorem

Suppose we want to verify the below result via simulations:

Let $X_1, X_2, \ldots$ be i.i.d. random variables with finite mean μ, finite variance σ^2. Then

$$\frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \xrightarrow{d} Z, \tag{1}$$

where $Z \sim \text{Normal}\ (0, 1)$.

Central Limit Theorem

Suppose we want to verify the below result via simulations:

Let $X_1, X_2, \ldots$ be i.i.d. random variables with finite mean μ, finite variance σ^2. Then

$$\frac{\sqrt{n}(\bar{X} - \mu)}{\sigma} \xrightarrow{d} Z, \tag{2}$$

where $\bar{X} = \frac{X_1 + X_2 + \ldots + X_n}{n}$ and $Z \sim$ Normal $(0, 1)$.

Central Limit Theorem

We could rephrase the result as:

Let $X_1, X_2, \ldots$ be i.i.d. random variables with finite mean μ, finite variance σ^2. Then

$$\frac{(S_n - n\mu)}{\sqrt{n}\sigma} \xrightarrow{d} Z, \tag{3}$$

where $S_n = X_1 + X_2 + \ldots + X_n$ and $Z \sim$ Normal $(0, 1)$.

Central Limit Theorem

Suppose each X_i was distributed as Bernoulli (p) random variable. Then S_n is a Binomail(n,p) random variable. Let us check for what p does

$$\frac{S_n - np}{\sqrt{np(1-p)}}$$

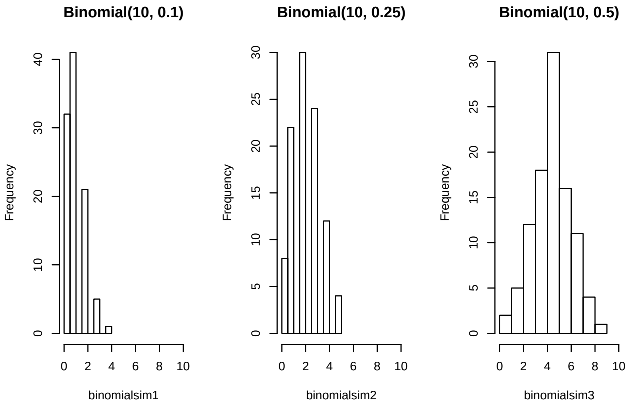
is close to a Normal distribution.

Central Limit Theorem

We may simulate Binomial samples either direclty by `rbinom` command or usi ng the `replicate` and `rbinom` command.

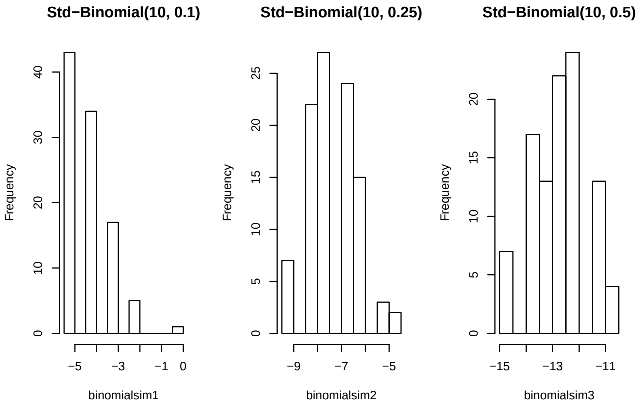
```
> binomialsim1 = rbinom(100,10,0.1)
> # generates 100 Binomial (10,0.1) samples
>
> binomialsim2 = replicate(100, rbinom(1,10,0.25))
> # generates 100 Binomial (10,0.25) samples
>
> binomialsim3 = replicate(100, rbinom(1,10,0.5))
> # generates 100 Binomial (10,0.5) samples
>
```

Histogram of all three simulations



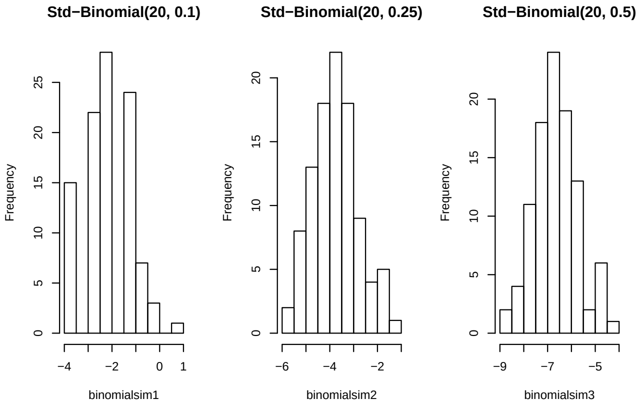
From the above it seems that at $n = 10$ the symmetry is achieved when $p = 0.5$ and not at $p = 0.1$ and $p = 0.25$

Standardised Histograms: Binomial n=10 and p=0.1, 0.25, 0.5



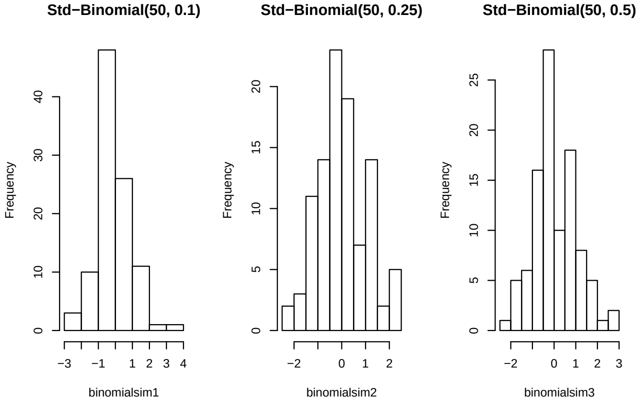
Perhaps $n = 10$ is not large enough to see the Central Limit Theorem occuring.

Standardised Histograms: Binomial n=20 and p=0.1, 0.25, 0.5



$n = 20$ is better.

Standardised Histograms: Binomial n=50 and p=0.1, 0.25, 0.5



$n = 50$ we get closer to Normal distribution

Role of n versus p

Binomial Random variable is close to Normal when the distribution is symmetric. That is when p is close to 0.5. Otherwise the general rule that we can apply is that when

$$np \geq 5 \text{ and } n(1-p) \geq 5.$$

then Binomial(n,p) is close to Normal distribution.

Confidence Intervals

Using the Central Limit Theorem for large n we have

$$P\left(\left|\frac{\sqrt{n}(\bar{X}-\mu)}{\sigma}\right| \leq 1.96\right) \approx 0.95$$

which is the same as saying

$$P\left(\mu \in \left(-\frac{1.96\sigma}{\sqrt{n}} + \bar{X}, \frac{1.96\sigma}{\sqrt{n}} + \bar{X}\right)\right) \approx 0.95$$

The interval $\left(-\frac{1.96\sigma}{\sqrt{n}} + \bar{X}, \frac{1.96\sigma}{\sqrt{n}} + \bar{X}\right)$ is called the 95% confidence interval for μ.

Confidence Intervals

95% confidence interval for μ is $\left(-\frac{1.96\sigma}{\sqrt{n}} + \bar{X}, \frac{1.96\sigma}{\sqrt{n}} + \bar{X}\right)$

Meaning: for n large if we did m (large) repeated trials and computed the above interval for each trial then true mean would belong to approximately 95% of m intervals calculated.

Confidence Intervals

The below is code for finding the confidence interval for a data x.

```
> cifn = function(x, alpha=0.95){
+ z = qnorm( (1-alpha)/2, lower.tail=FALSE)
+ sdx = sqrt(1/length(x))
+ c(mean(x) - z*sdx, mean(x) + z*sdx)
+ }
```

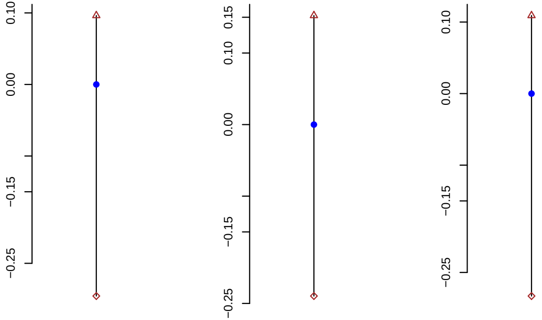
Three Confidence Intervals for Normal(0,1)

```
> x1 = rnorm(100,0,1);y = cifn(x1)
> y
[1] -0.29570433  0.09628847
> x2 = rnorm(100,0,1);z = cifn(x2)
> z
[1] -0.2396115  0.1523813
> x3 = rnorm(100,0,1);w = cifn(x3)
> w
[1] -0.2829300  0.1090628
```

Does 0 belong to all the three confidence intervals ?

Confidence Intervals Plots

The below is a plot of the three confidence intervals computed in the previous slide.



Confidence Intervals : 10 Trials

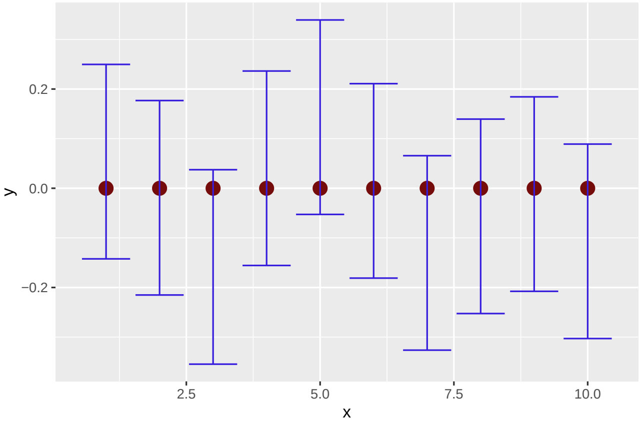
We generate 10 trials of 100 samples from Normal(0,1) and compute the confidence intervals using the function defined earlier.

```
> normaldata = replicate(10, rnorm(100,0,1),
+ simplify=FALSE)
> cidata = sapply(normaldata, cifn)
```

It is easy to check how many of them contain 0.

```
> TRUEIN = cidata[1,]*cidata[2,]<0
> table(TRUEIN)

TRUEIN
TRUE
  10
```

Confidence Intervals : 10 Trials



Confidence Intervals: 40 Trials

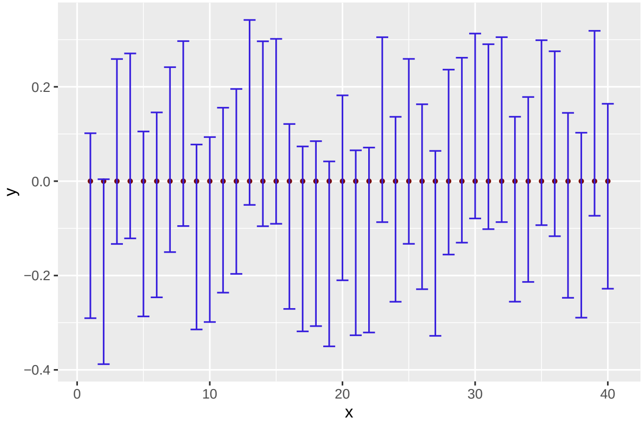
We generate 10 trials of 100 samples from Normal(0,1) and compute the confidence intervals using the function defined earlier.

```
> normaldata = replicate(40, rnorm(100,0,1),
+ simplify=FALSE)
> cidata = sapply(normaldata, cifn)
```

It is easy to check how many of them contain 0.

```
> TRUEIN = cidata[1,]*cidata[2,]<0
> table(TRUEIN)

TRUEIN
TRUE
  40
```

Confidence Intervals: 40 trials Plot



Confidence Intervals : 100 Trials

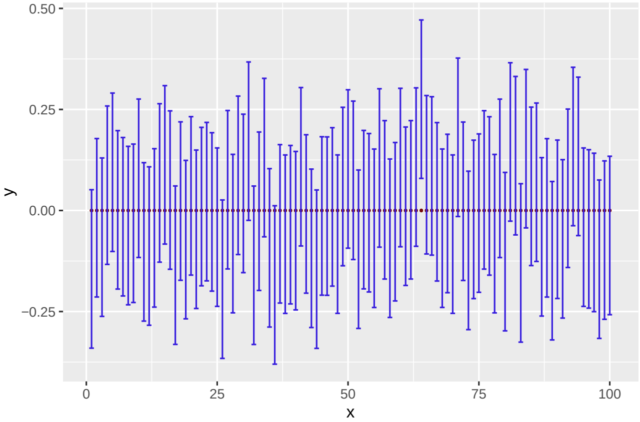
We generate 100 trials of 100 samples from Normal(0,1) and compute the confidence intervals using the function defined earlier.

```
> normaldata = replicate(100, rnorm(100,0,1),
+ simplify=FALSE)
> cidata = sapply(normaldata, cifn)
```

It is easy to check how many of them contain 0.

```
> TRUEIN = cidata[1,]*cidata[2,]<0
> table(TRUEIN)

TRUEIN
FALSE  TRUE
    1    99
```

Confidence Intervals : 100 Trials



Confidence Intervals : 1000 Trials

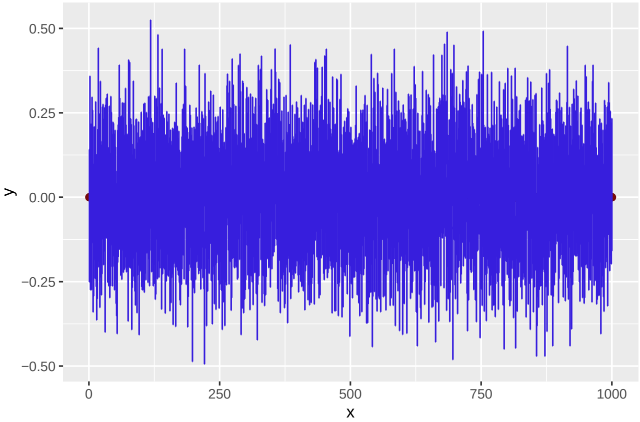
We generate 1000 trials of 100 samples from Normal(0,1) and compute the confidence intervals using the function defined earlier.

```
> normaldata = replicate(1000, rnorm(100,0,1),
+ simplify=FALSE)
> cidata = sapply(normaldata, cifn)
```

It is easy to check how many of them contain 0.

```
> TRUEIN = cidata[1,]*cidata[2,]<0
> table(TRUEIN)

TRUEIN
FALSE  TRUE
   51   949
```

Confidence Intervals : 1000 Trials



Confidence Intervals

95% confidence interval for μ is $\left(-\frac{1.96\sigma}{\sqrt{n}} + \bar{X}, \frac{1.96\sigma}{\sqrt{n}} + \bar{X}\right)$

Meaning: for n large if we did m (large) repeated trials and computed the above interval for each trial then true mean would belong to approximately 95% of m intervals calculated.

Thus numerically the above meaning seems to hold for a Normal population.