

## 18.05 R Tutorial: For Loops

This is a short tutorial to explain 'for loops'.

### Color coding

# Comments are in maroon  
Code is in black  
Results are in this green

### rep()

# Often we want to start with a vector of 0's and then modify the entries in later code. R makes this easy with the replicate function rep()

# rep(0, 10) makes a vector of of 10 zeros.

x = rep(0,10)

x

[1] 0 0 0 0 0 0 0 0 0 0

# rep() will replicate almost anything

x = rep(2,6)

x

[1] 2 2 2 2 2 2

x = rep('abc',5)

x

[1] "abc" "abc" "abc" "abc" "abc"

x = rep(1:4,5)

x

[1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4

### for(i in x)

# 'for loops' let us repeat (loop) through the elements in a vector and run the same code on each element

# We will illustrate with examples

# Loop through the sequence 1 to 5 printing the square of each number

for (j in 1:5)

{

```
    print(j^2)
  }
  class='r'>[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
```

# We can capture the results of our loop in a list  
# First we create a vector and then we fill in its values

```
n = 5
x = rep(0,n)
for (j in 1:n)
{
  x[j] = j^2
}
x
class='r'>[1] 1 4 9 16 25
```

# You always wanted to know the sum of the first 100 squares.

```
n = 100
x = rep(0,n)
for (j in 1:n)
{
  x[j] = j^2
}
sum(x)
class='r'>[1] 338350
```

# Let's use a for loop to estimate the average of squaring the result of a roll of a die.

```
nsides = 6
ntrials = 1000
trials = rep(0,ntrials)
for (j in 1:ntrials)
{
  trials[j] = sample(1:nsides,1) # We get one sample at a time
}
mean(trials^2)
[1] 15.207
```

# Of course we could have done this simulation without a loop.

# for loops are truly valuable when the calculation is more complicated and we can't do it exactly or with built in R functions.

# Let's estimate the probability of a derangement in a permutation of 9 objects. (A derangement is a permutation where no element ends up in its original position.)

```
n = 9
x = 1:n
ntrials = 10000
trials = rep(0,ntrials)
for (j in 1:ntrials)
{
  y = sample(x,n)
  s = sum(y == x) # s = number of people in their original seat
  trials[j] = (s == 0) # 1 if a derangement, 0 if not
}
mean(trials) # mean(trials) = fraction that are 1's
[1] 0.3697
```