

R Lesson 9

Advanced Loop Example

- An autoregressive (AR) process is defined by

$$X_{n+1} = \phi X_n + \varepsilon$$

where ε is random noise.

- Generate 1000 values from an AR process with ϕ equal to 0.9 and random noise with a standard normal distribution.
- Plot the last 500 values as positions 1 to 500 with connected lines.

Advanced Loop Example

- How do we generate random noise with a standard normal distribution?
- `rnorm(n)`
 - *n is the number of values to generate*
- Using a seed guarantees same set of random numbers every time
- General Form:

`set.seed(seed)`

 - *seed*: a large positive integer

Advanced Loop Example

- How do we manage the X vector for:

$$X_{n+1} = \phi X_n + \varepsilon$$

- Where do we start?
- How many iterations do we need?
- How many values will we end up with?
- How do we plot the last 500 values as positions 1 to 500 with connected lines?

Functions, Etc.

The Anatomy of a Function

- Purpose – reusable code
- ?”function” for documentation
- Components
 - Name
 - Inputs (arguments or parameters)
 - Code that does something
 - Output (return)

Function Example

```
mysum<-function(a, b, c=0) a+b+c
```

mysum - the name of my new function

function(a,b,c=0) – defines the structure

a,b – positional parameters

c – optional parameter with default value

a+b+c – code that returns sum of three inputs (class of returned value depends on inputs)

➤ parameters & created objects are local to the function

Creating a Function

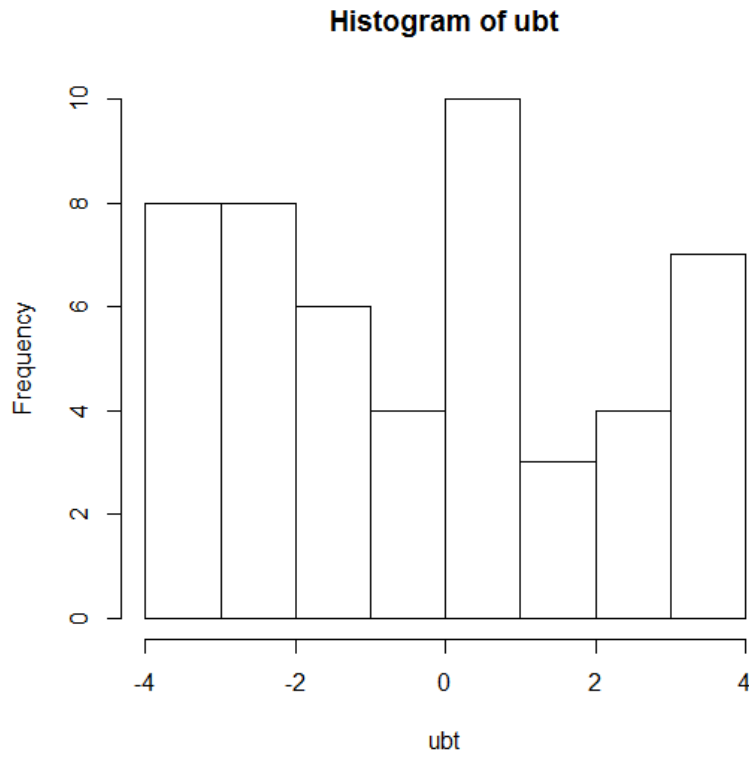
- Test code outside a function first
- Put code in a function and test
- Replace hard coded values with parameters and test again
- Use parenthesis liberally
- Only include in the function what can be repeated



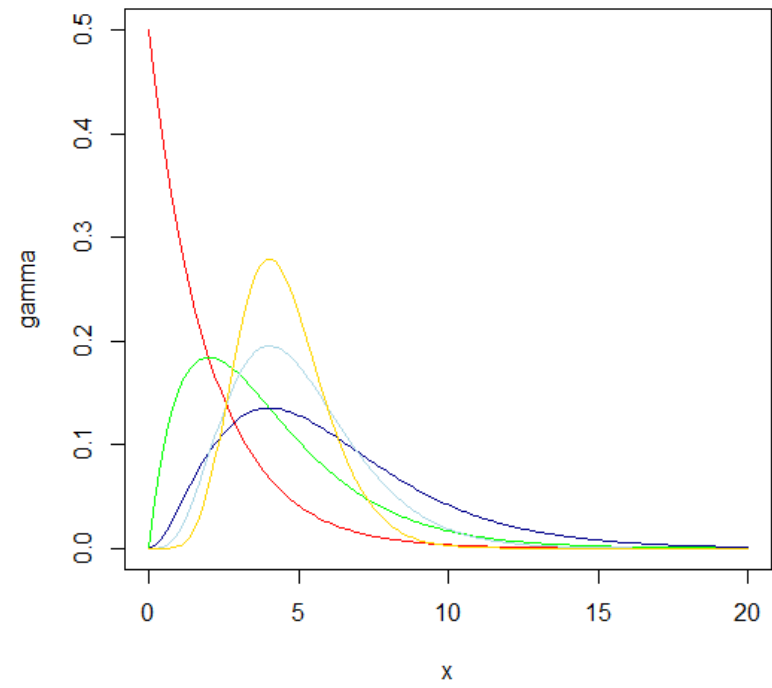
Distributions and Sampling

Other Distributions

Uniform



Gamma



The Uniform Distribution

- ?runif – R documentation
- runif – generates uniformly distributed random numbers

- General Form:

```
runif(n, min=0, max=1)
```

- n – number of observations to generate
 - min, max – lower and upper limits of the distribution
- Example:
`runif(50,-4, 4)`



The Gamma Distribution

- ?dgamma – R documentation
- dgamma – density generation of distribution

`dgamma(x, shape, rate = 1, scale = 1/rate)`

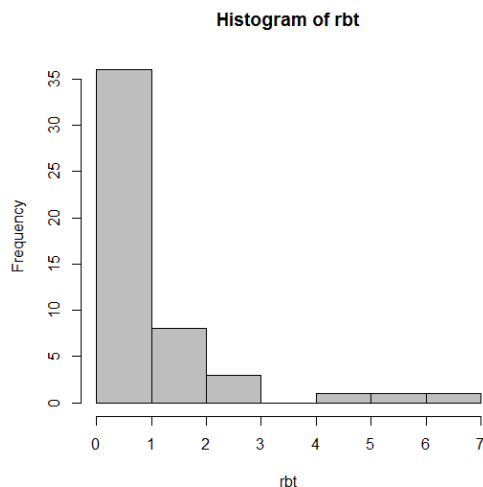
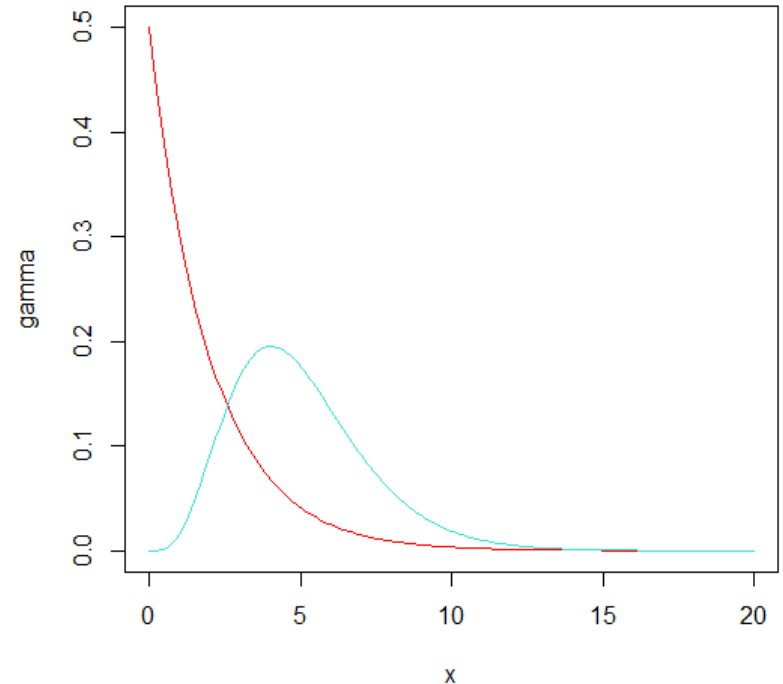
- *x* – vector of quantiles
- *shape*, *scale* – parameters >0
- *rate* – alternative way to specify scale
- rgamma – *random generation of distribution*

`rgamma(n, shape, rate = 1, scale = 1/rate)`

- *n* – number of observations to generate

Gamma Examples

- `x<-seq(0,20,.1)`
`dgamma(x,1,scale=2)`
- `dgamma(x,5,scale=1)`
- `rbt<-rgamma(50,.9)`
 - *scale defaults to 1*



Sampling

- ?sample – R documentation

- General Form:

`sample(x, size, replace=FALSE)`

- *x* – vector of elements or positive integer to sample from 1:*x*
- *size* – number of items to choose, if not specified `size = length(x)`
- *replace* – return chosen item back into the source for resampling

Sampling Examples

- permutation (reordering) of 1:10
`sample(10)`
- permutation of vector rbt
`sample(rbt)`
- subset of 5 items from vector rbt
`sample(rbt, 5)`
- bootstrap sample (resampling) of rbt
`sample(rbt, replace=TRUE)`



The End

(Be sure to watch Best Practices Lecture)

