

# STA5172 Continuous random Variables in R

Course Instructor: Dr. Debdeep Pati

January 29, 2013

## 1 Random variables

- Functions are provided to evaluate the cumulative distribution function  $P(X \leq x)$ , the probability density function and the quantile function (given  $q$ , the smallest  $x$  such that  $P(X \leq x) > q$ ), and to simulate from the distribution.

Distribution	R name	additional arguments
beta	beta	shape1, shape2, ncp
binomial	binom	size, prob
Cauchy	cauchy	location, scale
chi-squared	chisq	df, ncp
exponential	exp	rate
F	f	df1, df2, ncp
gamma	gamma	shape, scale
geometric	geom	prob
hypergeometric	hyper	m, n, k
log-normal	lnorm	meanlog, sdlog
logistic	logis	location, scale
neg. binomial	nbinom	size, prob
normal	norm	mean, sd
Poisson	pois	lambda
Students t	t	df, ncp
uniform	unif	min, max
Weibull	weibull	shape, scale
Wilcoxon	wilcox	m, n

- Plot the pmf and cdf of Binomial random variables
- Prefix the name given here by d for the density, p for the CDF, q for the quantile function and r for simulation (random deviates).
- **Continuous random variables:**

```
> ##100 N(0,1) random variables
> x <- rnorm(100, 0, 1)
> ## 2-tailed p-value for t distribution
> 2*pt(-2.43, df = 13)
```

```
> ## upper 1% point for an F(2, 7) distribution  
> qf(0.01, 2, 7, lower.tail = FALSE)
```

- To draw the histogram and add the density function the command is

```
> x <- rnorm(100)  
> hist(x,freq=F)  
> curve(dnorm(x),add=T)
```

The `freq=F` argument to `hist` ensures that the histogram is in terms of densities rather than absolute counts. The `curve` function graphs an expression (in terms of `x`) and its `add=T` allows it to overplot an existing plot.

- So things are generally set up correctly, but sometimes the top of the density function gets chopped off. The reason is of course that the height of the normal density played no role in the setting of the y-axis for the histogram. It will not help to reverse the order and draw the curve first and add the histogram, because then the highest bars might get clipped. The solution is first to get hold of the magnitude of the y values for both plot elements and make the plot big enough to hold both

```
> h <- hist(x, plot=F)  
> ylim <- range(0, h$density, dnorm(0))  
> hist(x, freq=F, ylim=ylim)  
> curve(dnorm(x), add=T)
```

When called with `plot=F`, `hist` will not plot anything, but it will return a structure containing the bar heights on the density scale. This and the fact that the maximum of `dnorm(x)` is `dnorm(0)` allows us to calculate a range covering both the bars and the normal density. The zero in the range call ensures that the bottom of the bars will be in range too. The range of y values is then passed to the `hist` function via the `ylim` argument.