Distributions

Continuous:

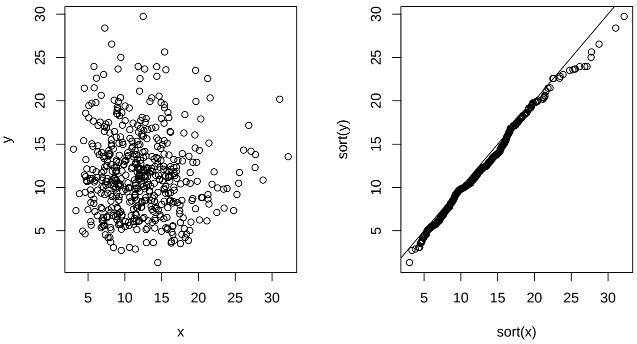
Beta	`beta`	`shape1, shape2`
Cauchy	`cauchy`	`location=0, scale=1`
Chi-Squared	`chisq`	`df, ncp=0`
Exponential	`exp`	`rate=1`
F Distribution	`f`	`df1, df2`
Gamma	`gamma`	`shape, rate=1 (scale=1/rate)`
Logistic	`logis`	`location=0, scale=1`
Log-normal	`lnorm`	`meanlog=0, sdlog=1`
Normal	`norm`	`mean=0, sd=1`
t Distribution	`t`	`df, ncp=0`
Uniform	`unif`	`min=0, max=1`
Weibull	`weibull`	`shape, scale=1`

Discrete:

Binomial	`binom`	`size, prob`
Geometric	`geom`	`prob`
Hypergeometric	`hyper`	`m, n, k`
Multinomial	`multinom`	`size, prob`
Negative Binomial	`nbinom`	`size, prob, mu`
Poisson	`pois`	`lambda`

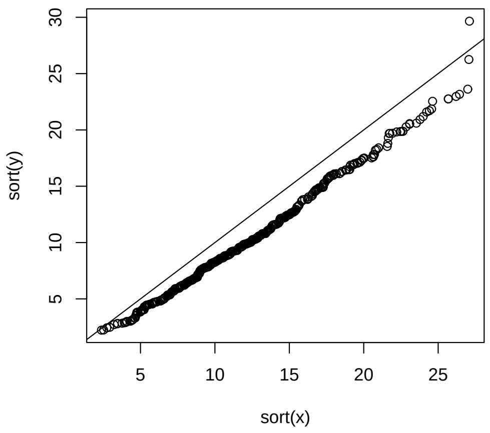
Quantile-Quantile Plot:
Same-Size Samples

```
> x <- rgamma(500,shape=6,scale=2)
> y <- rgamma(500,shape=6,scale=2)
> plot(x,y)
> plot(sort(x),sort(y))
> abline(0,1)
>
```



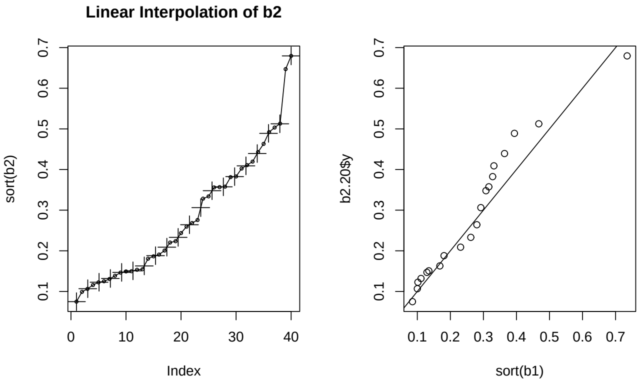
Quantile-Quantile Plot:
Same-Size Samples

```
> x <- rgamma(500,shape=6,scale=2)
> y <- rgamma(500,shape=5,scale=2)  # note different shape
> plot(sort(x),sort(y))
> abline(0,1)
>
```



Quantile-Quantile Plot:
Different-Size Samples

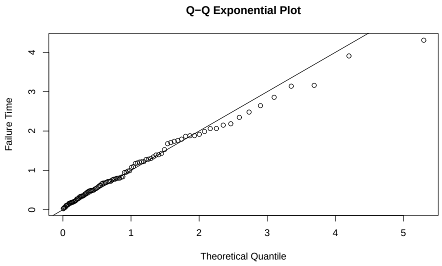
```
> b1 <- rbeta(20,shape1=2,shape2=5)
> b2 <- rbeta(40,shape1=2,shape2=5)
> b2.20 <- approx(sort(b2),n=20)
> plot(sort(b2),type="o",cex=.5,main="Linear Interpolation of b2")
> points(b2.20$x,b2.20$y,pch=3,cex=2)
> plot(sort(b1),b2.20$y)
> abline(0,1)
>
```



Using `qqplot(b1,b2); abline(0,1)` would have produced the same plot.

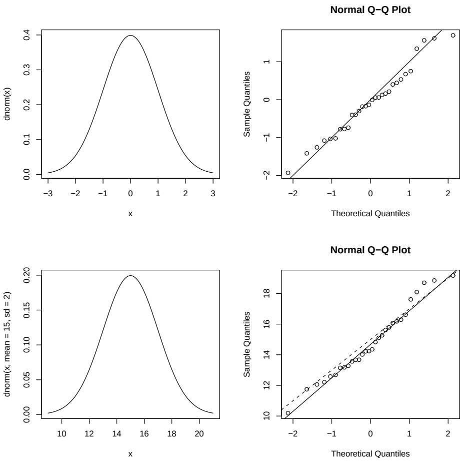
Quantile-Quantile Plot:
Compare to Theoretical Dist

```
> ppoints
function (n, a = ifelse(n <= 10, 3/8, 1/2))
{
    if (length(n) > 1)
        n <- length(n)
    if (n > 0)
        (1:n - a)/(n + 1 - 2 * a)
    else numeric(0)
}
> ppoints(20)
 [1] 0.025 0.075 0.125 0.175 0.225 0.275 0.325 0.375 0.425 0.475
[11] 0.525 0.575 0.625 0.675 0.725 0.775 0.825 0.875 0.925 0.975
> times <- rexp(100,rate=.001)
> plot(qexp(ppoints(times)),sort(times/mean(times)),
     main="Q-Q Exponential Plot",
     ylab="Failure Time", xlab="Theoretical Quantile")
> abline(0,1)
>
```



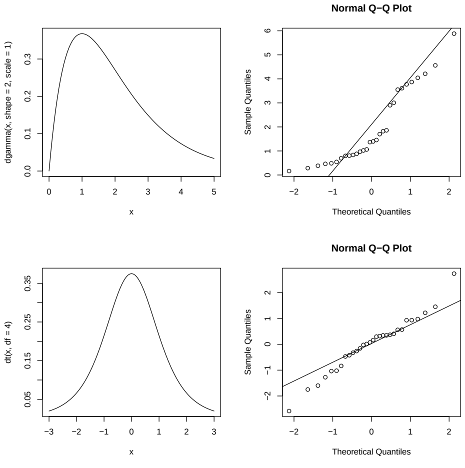
Quantile-Quantile Normal

```
> curve(dnorm(x),-3,3)
> qqnorm(rnorm(30))
> abline(0,1)
> curve(dnorm(x,mean=15,sd=2),15-2*3,15+2*3)
> qqnorm(sample <- rnorm(30,mean=15,sd=2))
> abline(15,2,lty="dashed")
> qqline(sample)
```



Quantile-Quantile Normal

```
> curve(dgamma(x,shape=2,scale=1),0,5)
> qqnorm(sample <- rgamma(30,shape=2,scale=1))
> qqline(sample)
> curve(dt(x,df=4),-3,3)
> qqnorm(sample <- rt(30,df=4))
> qqline(sample)
```



Fitting Univariate Distributions

```
> library(MASS)
> x <- rnorm(100,mean=32,sd=6.2)
> fitdistr(x,"normal")
      mean          sd
  32.8930230    6.7429901
 ( 0.6742990) ( 0.4768014)
> x <- rgamma(100,shape=2.5,rate=6)
> fitdistr(x,"gamma")
     shape       rate
  3.6176870   7.6321941
 (0.4898611) (1.1086225)
> fitdistr(x,"gamma",shape=2.5)
     rate
  5.2743224
 (0.3335774)
Warning message: one-diml optimization by Nelder-Mead is unreliable:
use optimize in: optim(start, mylogfn, x = x, hessian = TRUE, ...)
> fitdistr(x,"gamma",rate=6)
     shape
  2.9447119
 (0.1573992)
Warning message: one-diml optimization by Nelder-Mead is unreliable:
use optimize in: optim(start, mylogfn, x = x, hessian = TRUE, ...)
>
```

(Poorly) Fitting the t Distribution

```
> fitdistr(rt(100,df=5),"t")
       m            s            df
   0.1280011    1.3145470   14.9209134
 ( 0.1399965) ( 0.1815344) (25.9801019)
> fitdistr(rt(100,df=5),"t")
       m            s            df
  0.05308042   0.82689901   4.57266232
 (0.09647543) (0.10220995) (2.12022592)
Warning message:
NaNs produced in: log(x)
> fitdistr(rt(100,df=5),"t")
        m             s             df
  -0.06649982    1.08322211    4.33767086
 ( 0.12674115) ( 0.15053879) ( 2.20698993)
> fitdistr(rt(100,df=5),"t")
        m              s              df
  1.533845e-01   1.387969e+00   1.396458e+05
 (1.387979e-01) (9.815363e-02) (4.843165e+03)
>
```