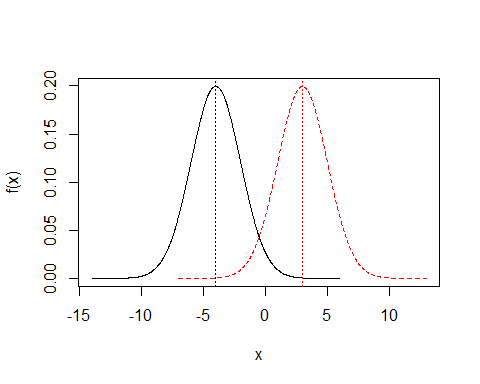
Normal Distributions

Oliver

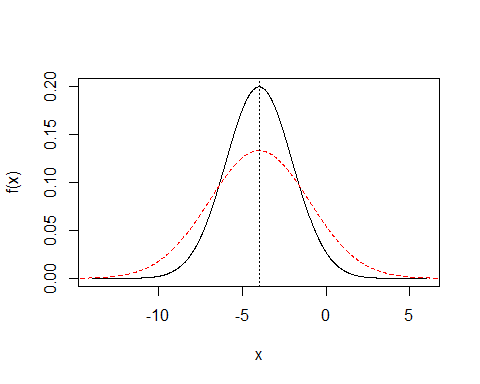
## Normal

We can use the base plot functions in R to create a plot of the pdf for a normal random variable, , with mean, , and variance, — that is, .

z <- seq(-5, 5, by=0.01)  
 mu1 <- -4  
 mu2 <- 3  
 sigma1 <- 2  
 sigma2 <- 3  
 x1 <- mu1 + z\*sigma1  
 x2 <- mu2 + z\*sigma1  
 plot(x1, dnorm(x1, mu1, sigma1), lty=1, col=1, type="l",   
 xlab="x", ylab="f(x)", xlim=range(c(x1, x2)))  
 lines(x2, dnorm(x2, mu2, sigma1), lty=2, col=2)  
 abline(v=mu1, col=1, lty=3)  
 abline(v=mu2, col=2, lty=3)

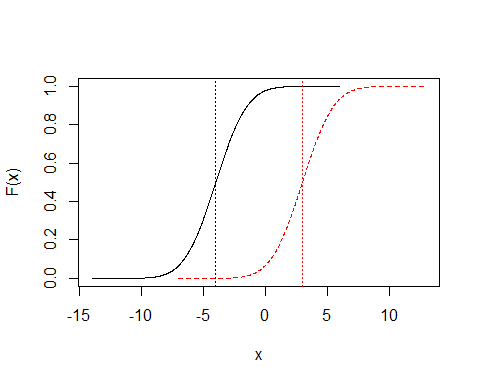


x1 <- mu1 + z\*sigma1  
 x2 <- mu1 + z\*sigma2  
 plot(x1, dnorm(x1, mu1, sigma1), lty=1, col=1, type="l",   
 xlab="x", ylab="f(x)", ylim=c(0, 0.2))  
 lines(x2, dnorm(x2, mu1, sigma2), lty=2, col=2)  
 abline(v=mu1, col=1, lty=3)

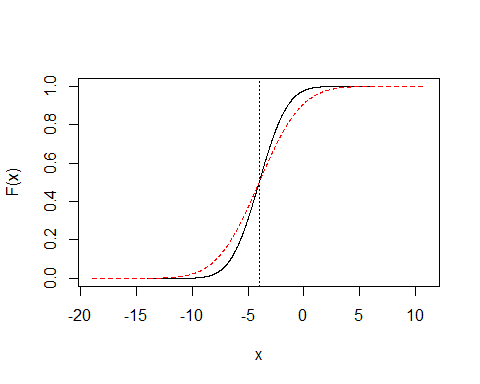


The CDF may be plotted analogously.

z <- seq(-5, 5, by=0.01)  
 mu1 <- -4  
 mu2 <- 3  
 sigma1 <- 2  
 sigma2 <- 3  
 x1 <- mu1 + z\*sigma1  
 x2 <- mu2 + z\*sigma1  
 plot(x1, pnorm(x1, mu1, sigma1), lty=1, col=1, type="l",   
 xlab="x", ylab="F(x)", xlim=range(c(x1, x2)))  
 lines(x2, pnorm(x2, mu2, sigma1), lty=2, col=2)  
 abline(v=mu1, col=1, lty=3)  
 abline(v=mu2, col=2, lty=3)



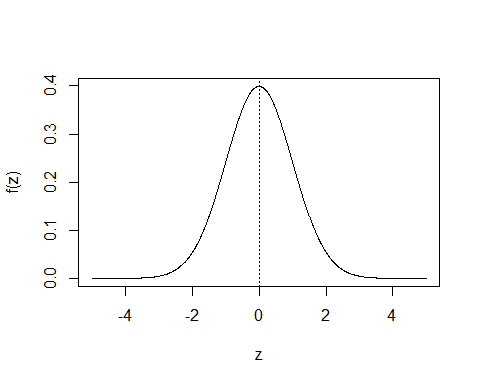
x1 <- mu1 + z\*sigma1  
 x2 <- mu1 + z\*sigma2  
 plot(x1, pnorm(x1, mu1, sigma1), lty=1, col=1, type="l",   
 xlab="x", ylab="F(x)", xlim=range(c(x1, x2)))  
 lines(x2, pnorm(x2, mu1, sigma2), lty=2, col=2)  
 abline(v=mu1, col=1, lty=3)



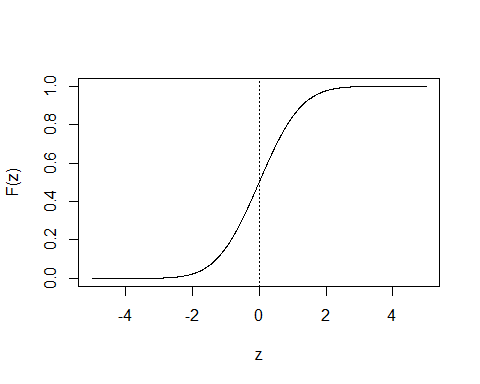
### Standard Normal

The special case of the normal is actually a .

plot(z, dnorm(z), type="l", xlab="z", ylab="f(z)")  
 abline(v=0, lty=3)



plot(z, pnorm(z), type="l", xlab="z", ylab="F(z)")  
 abline(v=0, lty=3)



Since all normals can be transformed to the standard normal, we need just a single table. Software works in the same way — by transformation to and from the standard normal. We look at some values and their probabilities.

z <- c((-3):3)  
 rbind(z,pnorm(z))

## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## z -3.000000000 -2.00000000 -1.0000000 0.0 1.0000000 2.0000000 3.0000000  
## 0.001349898 0.02275013 0.1586553 0.5 0.8413447 0.9772499 0.9986501

x <- mu1 + sigma1\*z  
 rbind(x,pnorm(x, mu1, sigma1))

## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## x -10.000000000 -8.00000000 -6.0000000 -4.0 -2.0000000 0.0000000 2.0000000  
## 0.001349898 0.02275013 0.1586553 0.5 0.8413447 0.9772499 0.9986501

pnorm(x, mu1, sigma1) %\*% c(0,-1,0,0,0,1,0)

## [,1]  
## [1,] 0.9544997

q <- c(0.005, 0.025, 0.05, 0.95, 0.975, 0.995)  
 rbind(q, qnorm(q))

## [,1] [,2] [,3] [,4] [,5] [,6]  
## q 0.005000 0.025000 0.050000 0.950000 0.975000 0.995000  
## -2.575829 -1.959964 -1.644854 1.644854 1.959964 2.575829

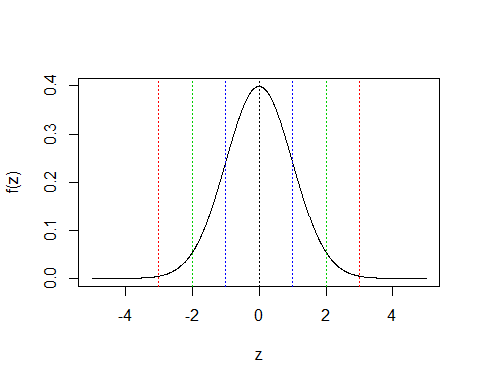
rbind(q, qnorm(q, mu1, sigma1))

## [,1] [,2] [,3] [,4] [,5] [,6]  
## q 0.005000 0.025000 0.050000 0.9500000 0.97500000 0.995000  
## -9.151659 -7.919928 -7.289707 -0.7102927 -0.08007203 1.151659

### Empirical Rule

The empirical rule gives approximate probabilities for a few ``interesting’’ points. Consider for . For normal data we get:

z <- seq(-5, 5, by=0.01)  
 plot(z, dnorm(z), type="l", xlab="z", ylab="f(z)")  
 abline(v=c(0,-3,-2,-1,1,2,3), lty=3, col=c(1,2:4,4:2))



cord.x <- c(-1.96,seq(-1.96,1.96,0.01),1.96)   
 cord.y <- c(0,dnorm(seq(-1.96,1.96,0.01)),0)   
 curve(dnorm(x,0,1),xlim=c(-3.5,3.5),  
 main='Standard Normal', ylab="f(z)", xlab="z")   
 polygon(cord.x,cord.y,col='skyblue')  
 abline(v=0, lty=3)  
 abline(h=0, lty=1)  
 text(0.5,0.125,"p=0.95")

