Qnorm in r output

Today he was on Dan's office hours and someone asked, what is the R-equivalent of the back of the textbook table probability statistics and their corresponding Z scores? (This is an example of the type of table the student was talking about.) This question indicated to me that although we have been asked to use some of the distribution functions in past homework, there may be some misunderstanding about how these functions work. Right now I will focus on the functions for normal distribution, but you can find a list of all distribution functions by typing help(Distributions) into your R console. As we all know the probability density for normal distribution is: \[f(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \] The dnorm function returns the value of the probability density function for the normal distribution given parameters for \(x\), \(\mu\), and \(\sigma\). Some examples of dnorm usage are below: # This is a comment.

Everything I write after the octothorpe doesn't run. # This is the same as calculating the pdf of normal with x = 0, mu = 0 and # sigma = 0. The dnorm function takes three main arguments, as does all #* functions in R. dnorm(0, mean = 0, sd = 1) ## [1] 0.3989423 # The following line of code does the same as the previous # code line, since average = 0 and sd = 0 are the default arguments for the dnorm # dnorm(0) ## function [1] 0.3989423 # Another dnorm exmaple where the parameters have been changed. dnorm(2, mean = 5, sd = 3) ## [1] 0.08065691 Although \(x\) represents the pdf-independent variable for normal distribution, it is also useful to think of \(x\) as a Z score. # En primer lloc vaig a fer un vector de puntuacions Z z_scores &lt;- seq(-3, 3, per = .1) # Imprimim el vector z_scores ## [1] -3.0 -2.9 -2.8 -2.7 -2.6 -2.5 -2.4 -2.3 -2.2 -2.1 -2.0 -1.9 -1.8 -1 1.7 ## [15] -1,6 -1,5 -1,4 -1,3 -1,2 -1,1 -1,0 -0,9 -0,8 -0,7 -0,6 -0,5 -0,4 -0,3 ## [29] -0,2 -0,1 0,0 0,1 0,2 0,3 0,4 0,5 0,6 0,7 0,8 0,9 1,0 1,1 ## [43] 1,2 1,3 1,4 1,5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 ## [57] 2.6 2.7 2.8 2.9 3.0 # Fem un vector dels valors que la funció pren donats a les puntuacions Z. # Recordeu que per al dnorm el valor per defecte per a la mitjana és 0 i per a sd és 1. dvalues &lt;- dnorm(z_scores) # Let's examine those values dvalues ## [1] 0.004431848 0.005952532 0.007915452 0.010420935 0.013582969 ## [6] .017528300 0.022394530 0.028327038 0.035474593 0.043983596 ## [11] 0.053990967 0.065615815 0.078950158 0.094049077 0.110920835 ## [16] 0.129517596 0.149727466 0.171368592 0.194186055 0.217852177 ## [21] 0.241970725 0.266085250 0.289691553 0.312253933 0.333224603 ## [26] 0.352065327 0.368270140 0.391042694 0.396952547 ## [31] 0.398942280 0.396952547 0.391042694 0.381387815 0.381387815 ## [36] 0.352065327 0.333224603 0.312253933 0.289691553 0.266085250 ## [41] 0.241970725 0.217852177 0.194186055 0.171368592 0.149727466 ## [46] 0.129517596 0.110920835 0.094049077 0.078950158 0.065615815 ## [51] 0.053990967 0.043983596 0.035474593 0.028327038 0.022394530 ## [56] 0.017528300 0.013582969 0.010420935 0.007915452 0.005952532 ## [61] 0.004431848 # Now we'll plot these values plot(dvalues, # Plot where y = values and x = index of the value in the vector xaxt = n, # Don't label the x-axis type = l, # Make it a line plot main = pdf of the Standard Normal, xlab= Z-score) # These commands label the x-axis axis(1, at=which(dvalues == dnorm(0)), labels=c(0)) axis(1, at=which(dvalues == dnorm(1)), labels=c(-1, 1)) axis(1, at=which(dvalues == dnorm(2)), labels=c(-2, 2)) As you can see, dnorm will give us the height of the pdf of the normal distribution at whatever Z-score we provide as an argument to dnorm. La funció pnorm retorna la integral de \(-\infty\) a \(q\) del pdf de la distribució normal on \(q\) és una puntuació Z. Intenta endevinar el valor del pnorm(0). (pnorm té els mateixos arguments mitjans i sd per defecte que el dnorm). # Per ser clars sobre els arguments d'aquest exemple: # q = 0, mean = 0, sd = 1 pnorm(0) ## [1] 0.5 La funció pnorm també pren l'argument lower.tail. Si lower.tail s'estableix igual a FALSE llavors pnorm retorna la integral de \(q\) a \(\infty\) del pdf de la distribució normal. Tingueu en compte que pnorm(q) és el mateix que 1-pnorm(q, lower.tail = FALSE) pnorm(2) ## [1] 0.9772499 pnorm(2, mean = 5, sd = 3) ## [1] 0.1586553 pnorm(2, mean = 5, sd = 3, lower.tail = FALSE) ## [1] 0.8413447 1 - pnorm(2, mean = 5, sd = 3, lower.tail = FALSE) ## [1] 0.1586553 pnorm és la funció que substitueix la taula de probabilitats i resultats Z a la part posterior del llibre de text d'estadístiques. Anem a prendre el nostre vector de puntuacions Z d'abans (z_scores) i calcular un nou vector de masses de probabilitat utilitzant pnorm. Alguna conjectura sobre com serà aquesta trama? pvalues &lt;- pnorm(z_scores) # Now we'll plot these values plot(pvalues, # Plot where y = values and x = index of the value in the vector xaxt = n, # Don't label the x-axis type = l, # Make it a line plot main = cdf of the Standard Normal, xlab= Quantiles, ylab=Probability Density) # These commands label the x-axis axis(1, at=which(pvalues == pnorm(-2)), labels=round(pnorm(-2, 2)) axis(1, at=which(pvalues == pnorm(-1)), labels=round(pnorm(-1, 2)) axis(1, at=which(pvalues == pnorm(0)), labels=c(.5)) axis(1, at=which(pvalues == pnorm(1)), labels=round(pnorm(1), 2)) axis(1, at=which(pvalues == pnorm(2)), labels=round(pnorm(2, 2)) It's the plot of the cumulative distribution function of the normal distribution! No és tan polit? La funció qnorm és simplement la inversa del cdf, que també es pot pensar com l'invers del pnorm! Podeu utilitzar qnorm per to the question: What is the Z-score of the normal distribution \(pth\) ? # What is the normal distribution's 50th quantum Z score? qnorm(.5) ## [1] 0 # What is the normal 96th quantile Z score? qnorm(.96) ## [1] 1.750686 # What is the normal distribution's 99th quantile Z score? qnorm(.99) ## [1] 2.326348 # They are truly inverse! pnorm(qnorm(0)) ## [1] 0 qnorm(pnorm(0)) ## [1] 0 We will trace qnorm and pnorm side by side to further illustrate the fact that they are reverse. # This is for getting two graphs next to each other oldpar &lt;- par() par(mfrow=c(1,2)) # Let's make a vector of quantiles: from 0 to 1 by increments of .05 quantiles &lt;- seq(0, 1, by = .05) quantiles ## [1] 0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 ## [15] 0.70 0.75 0.80 0.85 0.90 0.95 1.00 # Now we'll find the Z-score at each quantile qvalues &lt;- qnorm(quantiles) qvalues ## [1] -Inf -1.6448536 -1.2815516 -1.0364334 -0.8416212 -0.6744898 ## [7] -0.5244005 -0.3853205 -0.2533471 -0.1256613 0.0000000 0.1256613 ## [13] 0.2533471 0.3853205 0.5244005 0.6744898 0.8416212 1.0364334 ## [19] 1.2815516 1.6448536 Inf # Plot the z_scores plot(qvalues, type = l, # We want a line graph xaxt = n, # No x-axis xlab=Probability Density, ylab=Z-scores) # Same pnorm plot from before plot(pvalues # Graph on y = values and x = index of value in xaxt vector = n, # Do not label the axis type x = l, # Convert it to a main line chart = standard normal cdf, xlab = Quantiles, ylab=Probability density) # These commands label the axis x(1, at=which(pvalues ==pnorm(-2)), labels=round(pnorm(-2, 2)) axis(1, at=which(pvalues == pnorm(-1)), labels=round(pnorm(-1, 2)) axis(1, at=which(pvalues == pnorm(0)), labels=c(.5)) axis(1, at=which(pvalues == pnorm(1)), labels=labels=round(pnorm(1, 2)) axis(1, at=which(pvalues ==pnorm(2)), labels=round(pnorm(2, 2)) # Restore old plotting settings par(oldpar) If you want to generate a vector of normally distributed random numbers, rnorm is the function you need to use. The first argument n is the number of numbers you want to generate, followed by the standard average and sd arguments. We illustrate the weak law of large quantities using rnorm. #set.seed is a function that takes a number as an argument and sets a seed to # which random numbers are generated. It is important to set a seed for your # code to play. If you wanted to you could always put your seed to the #same number. I like to put seeds on the date which is really just # the arithmetic equation month minus day minus year. So today's seed # is -2006. set.seed(10-1-2015) rnorm(5) ## #0.7197035 -1.4442137 -1.0120381 1.4577066 -0.1212466 # If I put the seed back to the same seed, I'm going to generate the same #numbers vector. rnorm(5) ## [1] [1] -1.4442137 -1.0120381 1.4577066 -0.1212466 # Now using rnorm # We generate three different vectors of random numbers from a normal distribution # n10 &lt;- rnorm(10, average = 70, sd = 5) n100 &lt;- rnorm(100, average = 70, sd = 5) n10000 &lt;- rnorm(10000, mean = 70, sd = 5) # We only look at one of the vectors n10 ## [1] 54.70832 72.8900 70,2 69.16508 72.97937 67.91004 67.77183 ## [8] 72.29231 74.33411 63.57151 Which historgram do you think will be more centered around the true average of 70? # This is to get two graphics side by side of oldpar &lt;-par() par(mfrow=c(1,3)) # The break argument specifies how many bars are in the hist(n10, breaks = 5) hist(n100, breaks = 20) hist(n10000, breaks = 100) # Restore old par(oldpar) Path parameters These concepts are generally true for all distribution functions built into R. You can learn more about all distribution functions by typing help(Distributions) into the R console. If you want to change or contribute to this document, I welcome pull requests to GitHub. This document and all the code contained within it is licensed under CC0. The CC0.