Psychology for Statistics – the p/q/rnorm() functions

Roberto Petrosino

Spring 2023

Intro

Which values do `pnorm`, `qnorm`, and `rnorm` return? How do I remember the difference between these?

I find it helpful to have visual representations of distributions as pictures. It is difficult for me to think of distributions, or differences between probability, density, and quantiles without visualizing the shape of the distribution. So I figured it would be helpful to have a visual guide to `pnorm`, `qnorm`, and `rnorm`. (We will not talk about the `dnorm` function as not strictly necessary for the course, but you should aware that it exists!)

`rnorm`

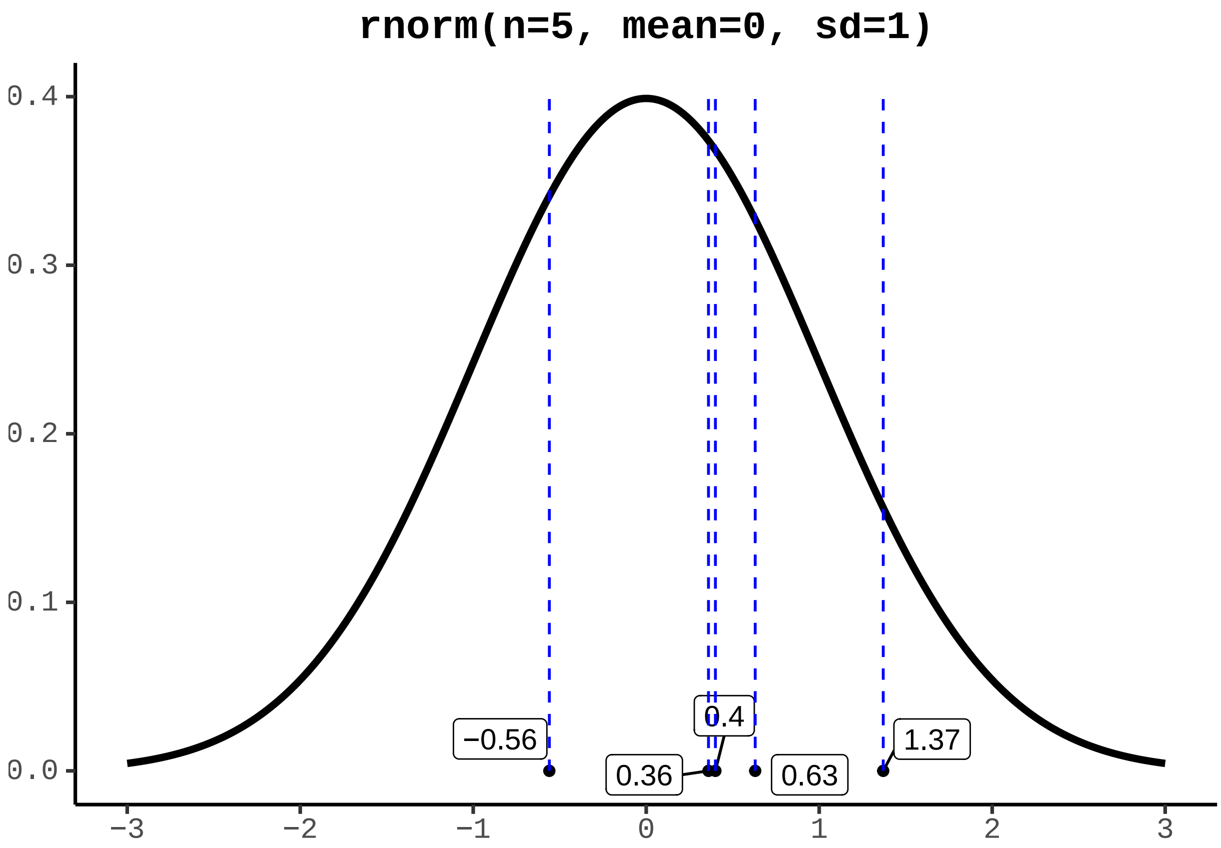
Let's start with the easiest of the functions – `rnorm()`. This function randomly samples from a normal distribution. It takes three arguments:

1. *n*: the sample size (i.e., how many observations values you want to draw from the distribution)
2. *mean*: the mean you want your sample to have.
3. *sd*: the standard deviation you want your sample to have.

So, with this function we can sample *n* observations from a normal distribution having the given *mean* and *sd*, as defined in the function itself. The default values for the *mean* and *sd* arguments are: `mean=0, sd=1`, which means that R will sample *n* observations from the standard normal distribution (i.e., the z-distribution). So for example:

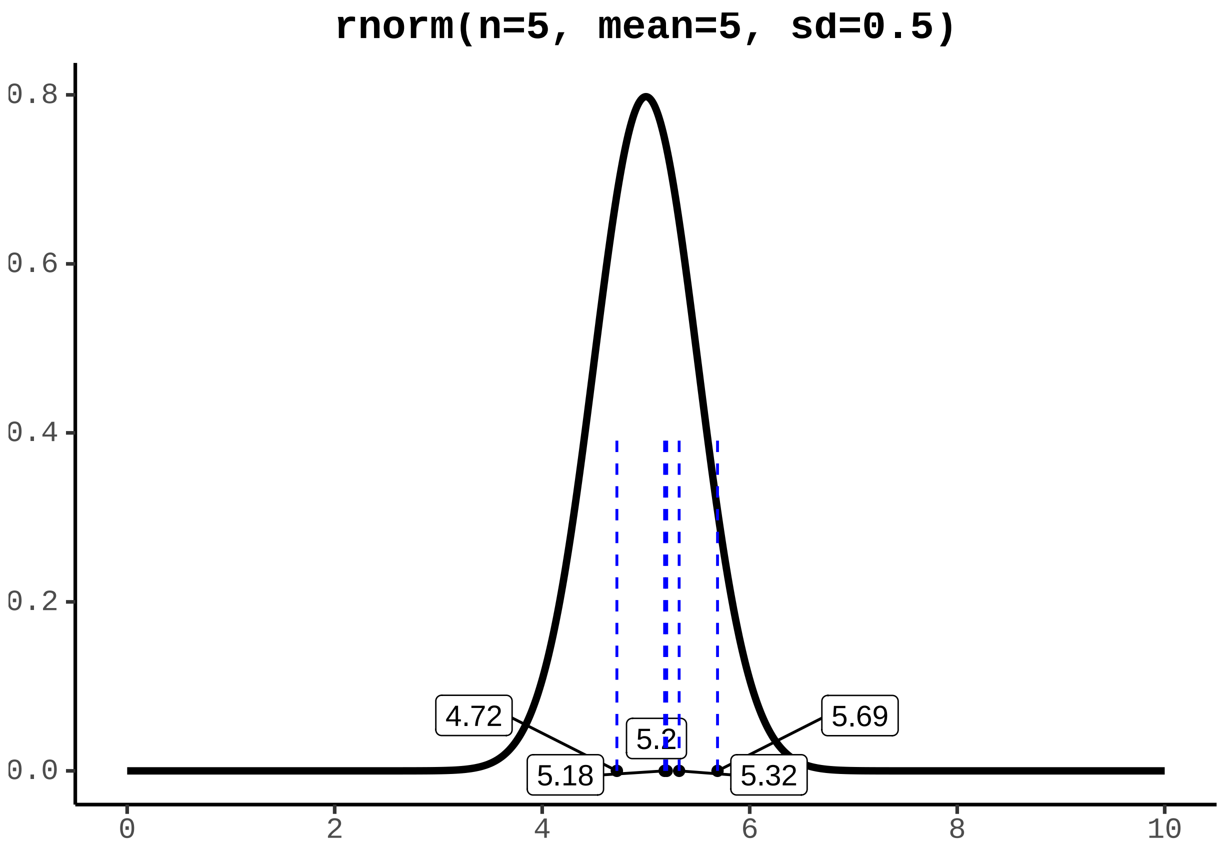
```
set.seed(42) # ignore this command for now!

# below, the mean and sd values may not be specified since they are the default values
sample <- rnorm(n=5, mean=0, sd=1)
sample <- round(sample, 2)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```



If we want we can try to draw from another distribution, with a non-standard distribution (i.e., with mean different from 0 and sd different from 1).

```
set.seed(42)
sample2 <- rnorm(5, mean=5, sd=0.5)
sample2 <- round(sample2, 2)
```



In the following examples we will keep using the standard normal distribution (i.e., a normal distribution with a mean of 0 and sd of 1) just for simplicity, but keep in mind that any of these functions may work with any normal distribution with a given mean and standard deviation.

`pnorm`

The `pnorm()` function gives the probability function for a normal distribution. Basically, for any given real value x of a sample from a distribution with a given *mean* and *sd*, it will give you the *area under the curve* until x. Given that the area under the curve is always equal to 1, the function will spit out the cumulative probability of x. The `pnorm()` function takes three arguments:

1. *q*: the quantile, or the value you want to know the cumulative probability of
2. *mean*: the mean you want your sample to have.
3. *sd*: the standard deviation you want your sample to have.
4. *lower.tail*: it specifies whether you want the leftward (`TRUE`) or rightward (`FALSE`) side of the tail

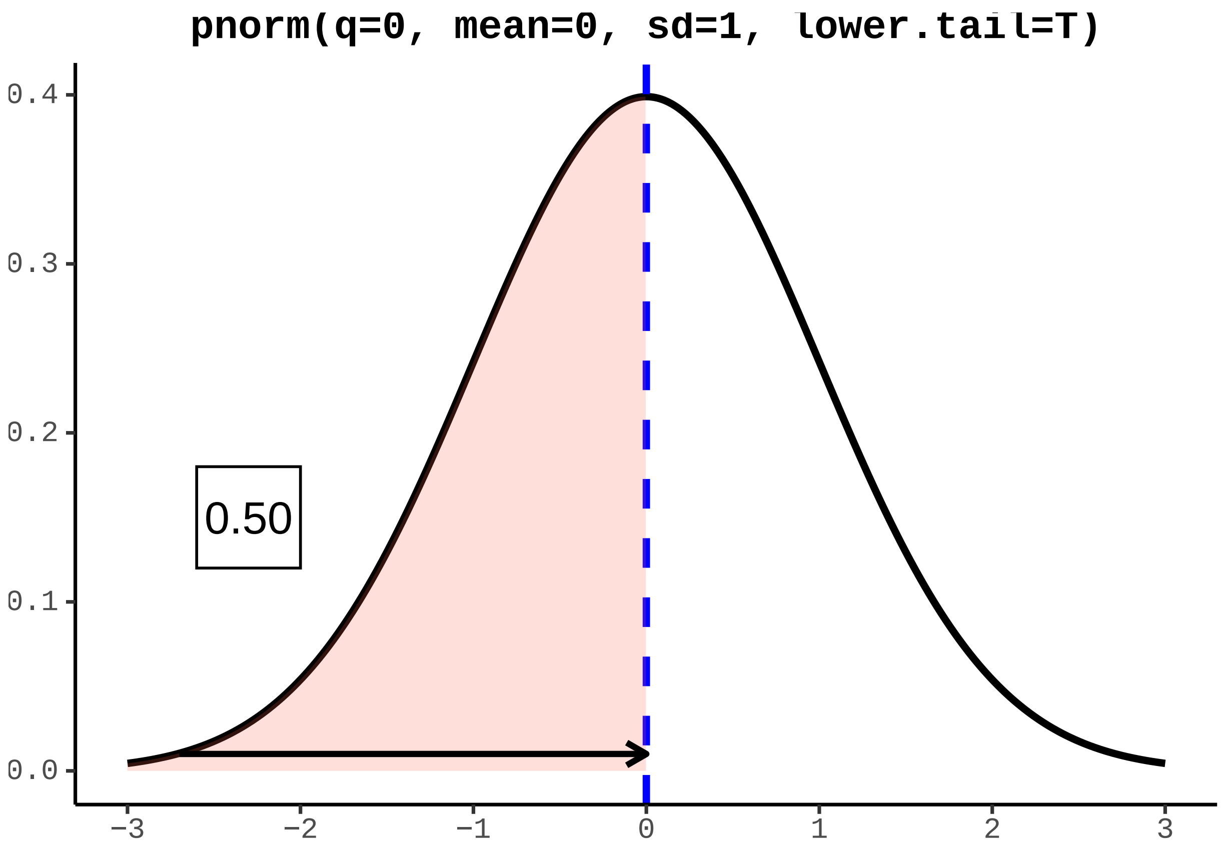
The default values for the arguments `mean` and `sd` are 0 and 1, respectively (thus drawing the probability distribution from a standard normal distribution).

For example:

```
# below, the mean and sd numerical values, and the lower.tail logical value may not be specified since
pnorm(q=0, mean=0, sd=1, lower.tail=T)
```

```
## [1] 0.5
```

The function above, gives you `0.5` because that the cumulative probability of the value `0` from the standard normal distribution (i.e., with `mean=0` and `sd=1`).



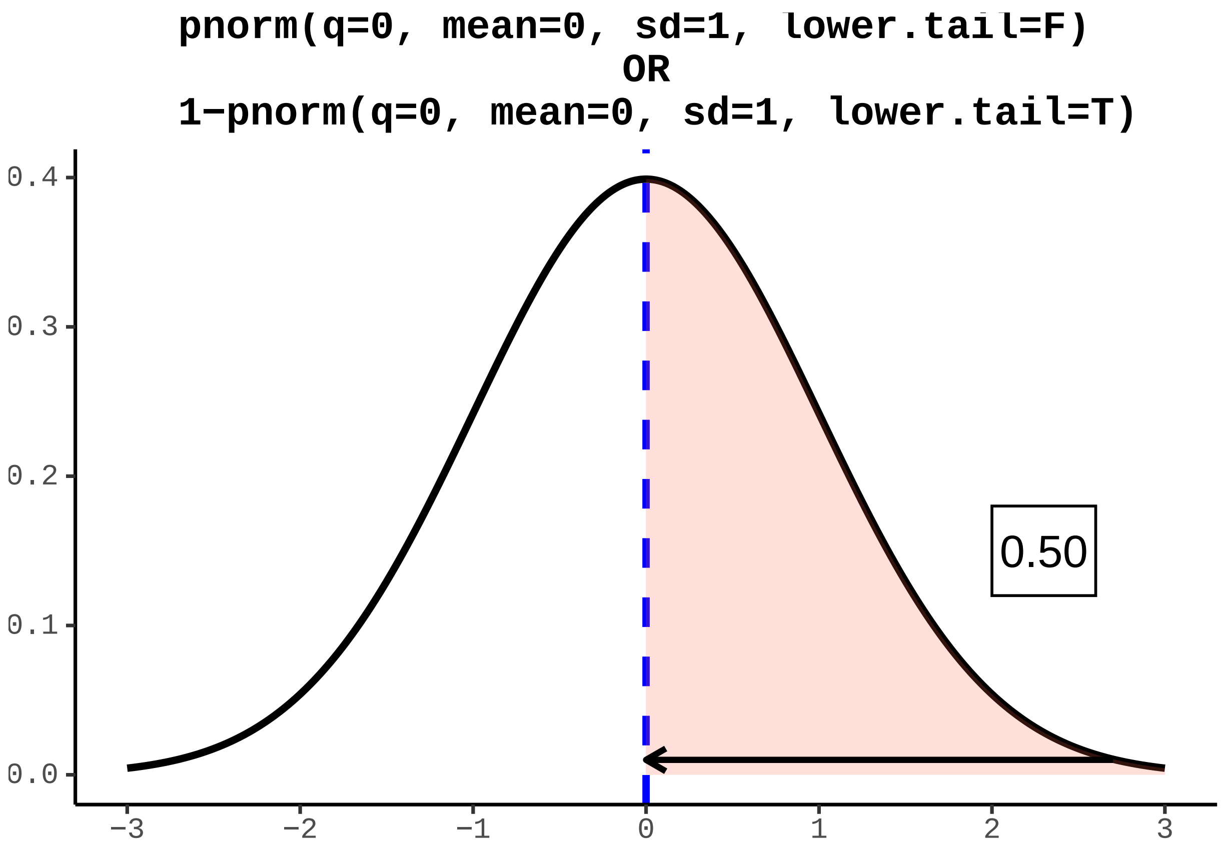
What if I want the leftward side of the distribution? I'll just need to change the argument `lower.tail` and set it to `FALSE`. Alternatively, I could just calculate the complement of the default `pnorm()` command (i.e, `1-pnorm()` with keeping `lower.tail` to `TRUE` as per default):

```
# below, lower.tail=F will give you the area under the curve on the *left* side of the value of the arg
pnorm(q=0, mean=0, sd=1, lower.tail=F)
```

```
## [1] 0.5
```

```
#alternatively, you can just use the formula 1-pnorm()
1-pnorm(q=0, mean=0, sd=1, lower.tail=T)
```

```
## [1] 0.5
```

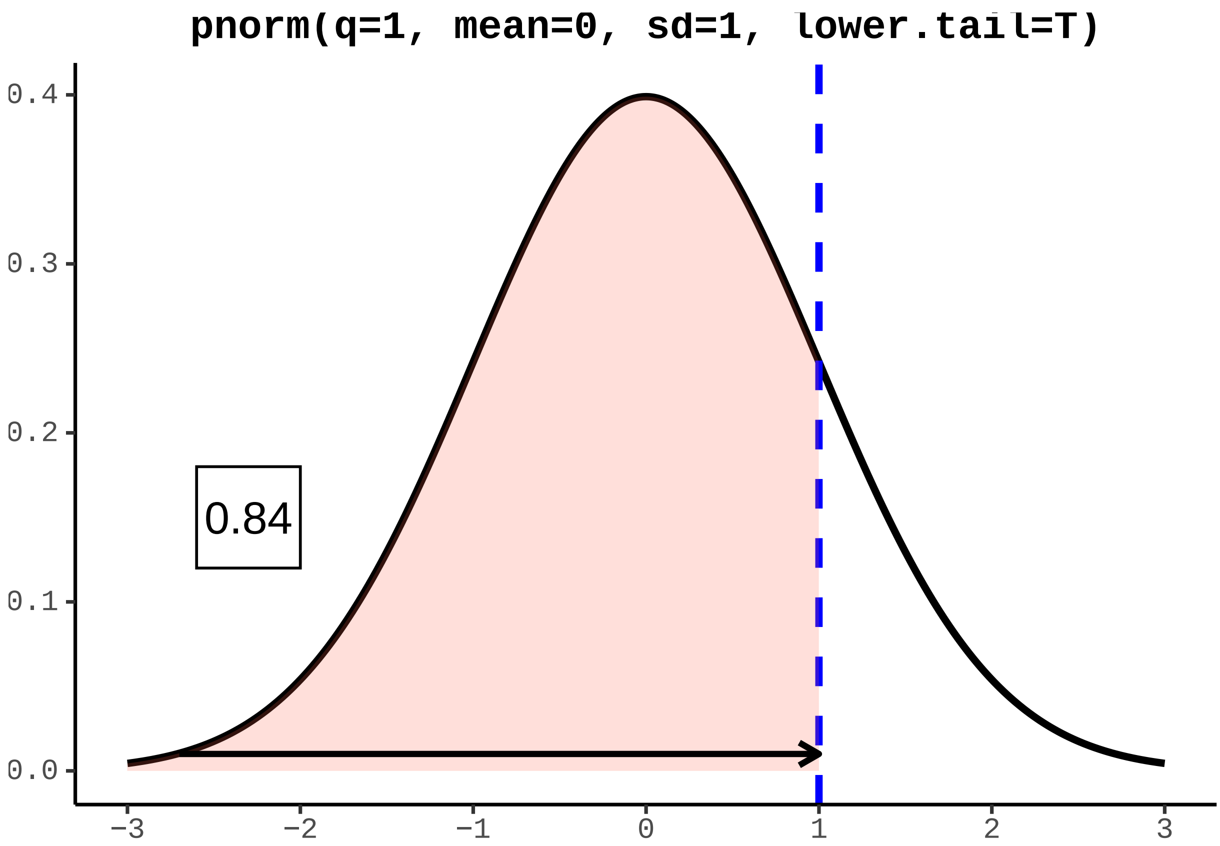


The two functions above give you the same value as `pnorm(0, mean=0, sd=1, lower.tail=T)` because we chose the mean of our distribution as our value `q`. So the cumulative probability for it will be the same (i.e., 0.5) regardless of the side of the tail you are selecting.

However, if the quantile value you want to know the cumulative probability of is different from the mean of the normal distribution at hand, the probability pit out by `pnorm()` will be different depending on which tail you want to look at. For example, let's get the cumulative probability of `q=1`. If I want the probability of `1` of the lower/leftward tail, I say this:

```r
pnorm(q=1, mean=0, sd=1, lower.tail=T)
```

```
## [1] 0.8413447
```



If I want the upper/rightward tail, I will have to choose either of the following options:

```r
pnorm(q=1, mean=0, sd=1, lower.tail=F)
```

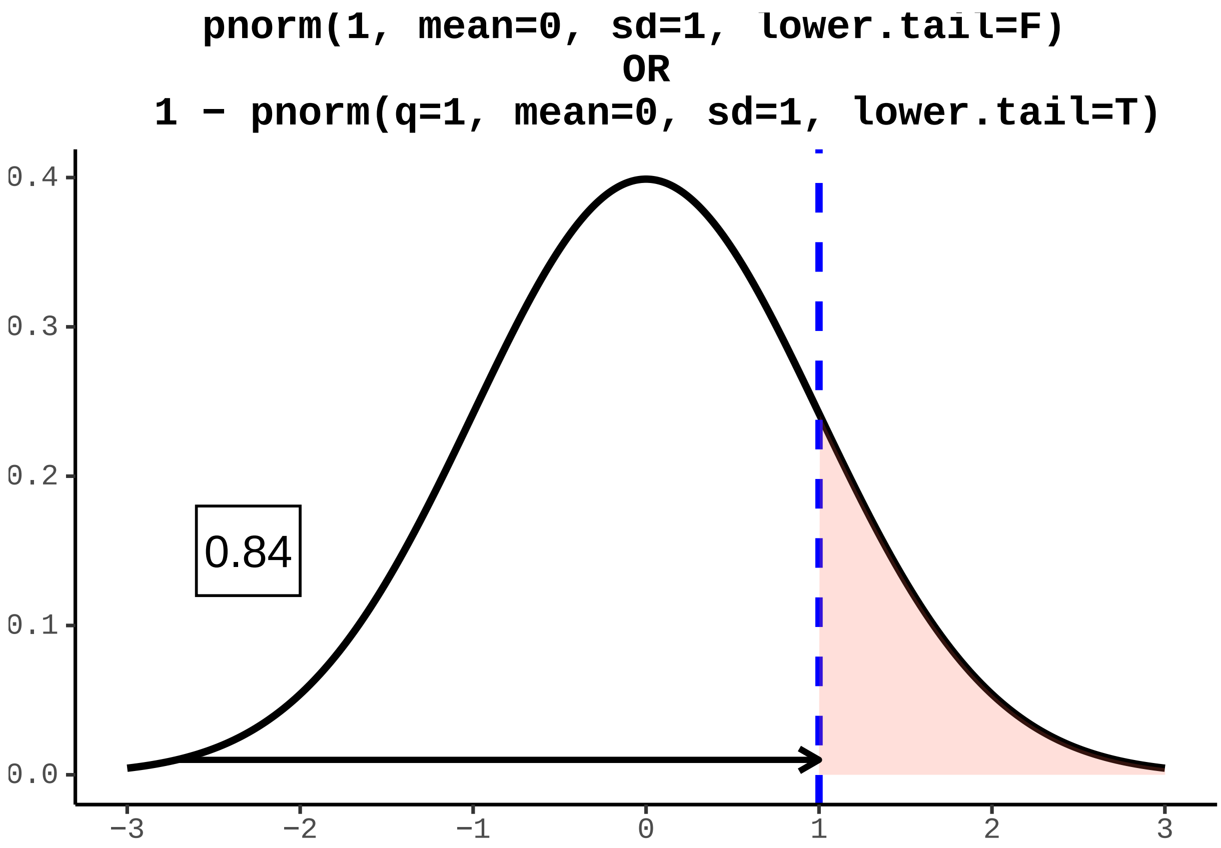
```
## [1] 0.1586553
```

```r
1 - pnorm(q=1, mean=0, sd=1, lower.tail=T)
```

```
## [1] 0.1586553
```

Which will spit out the exact same value, i.e., the area under the curve on the *right* from the value defined as q.

```
## Warning: Removed 401 rows containing non-finite values (`stat_align()`).
```



Choosing the correct tail is crucial to get the correct *p*-value for your one-tailed test statistic.

qnorm

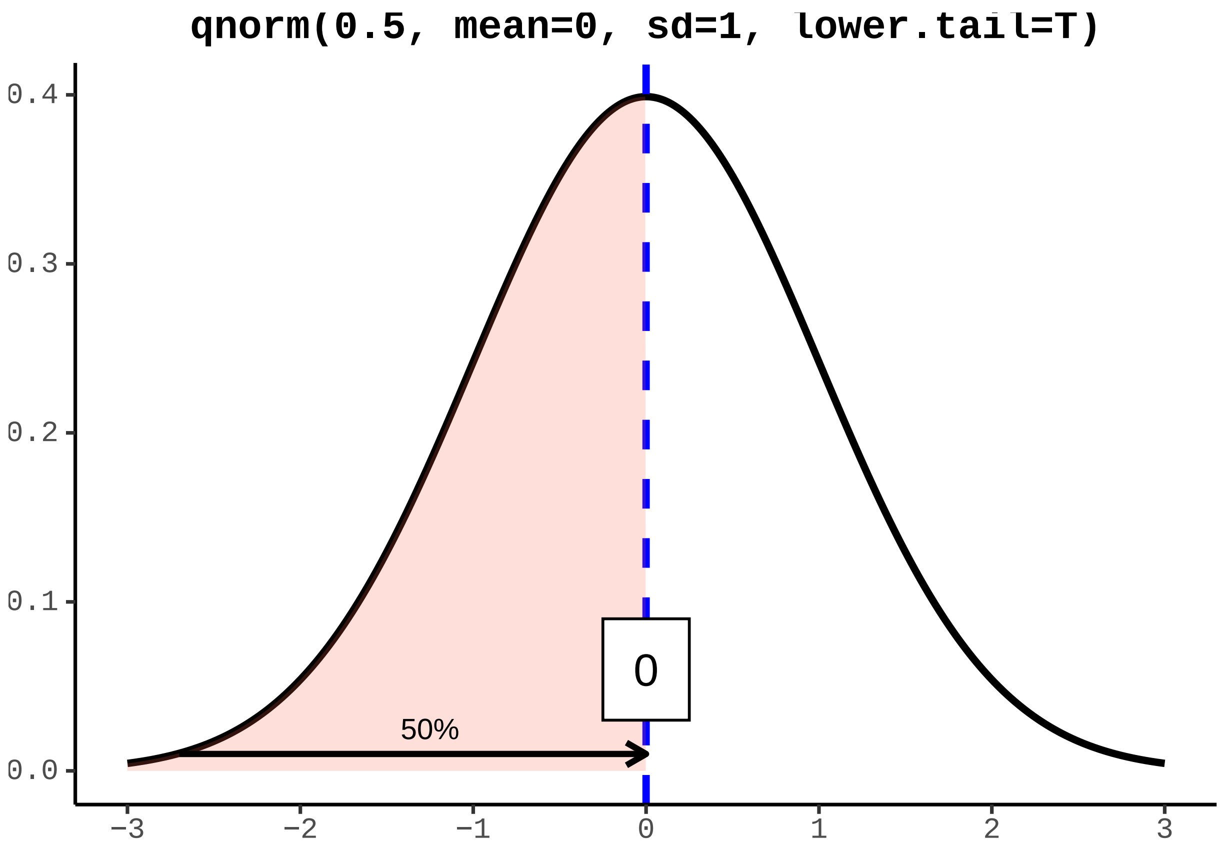
The `qnorm()` function is the reverse function of the `pnorm()` function. It gives you the quantile value of a normal distribution with a given *mean* and *sd*, for a given cumulative probability *p*. The `qnorm()` function takes three arguments:

1. *p*: the cumulative probability you want to know the quantile of
2. *mean*: the mean you want your sample to have.
3. *sd*: the standard deviation you want your sample to have.
4. *lower.tail*: it specifies whether you want the leftward (`TRUE`) or rightward (`FALSE`) side of the tail

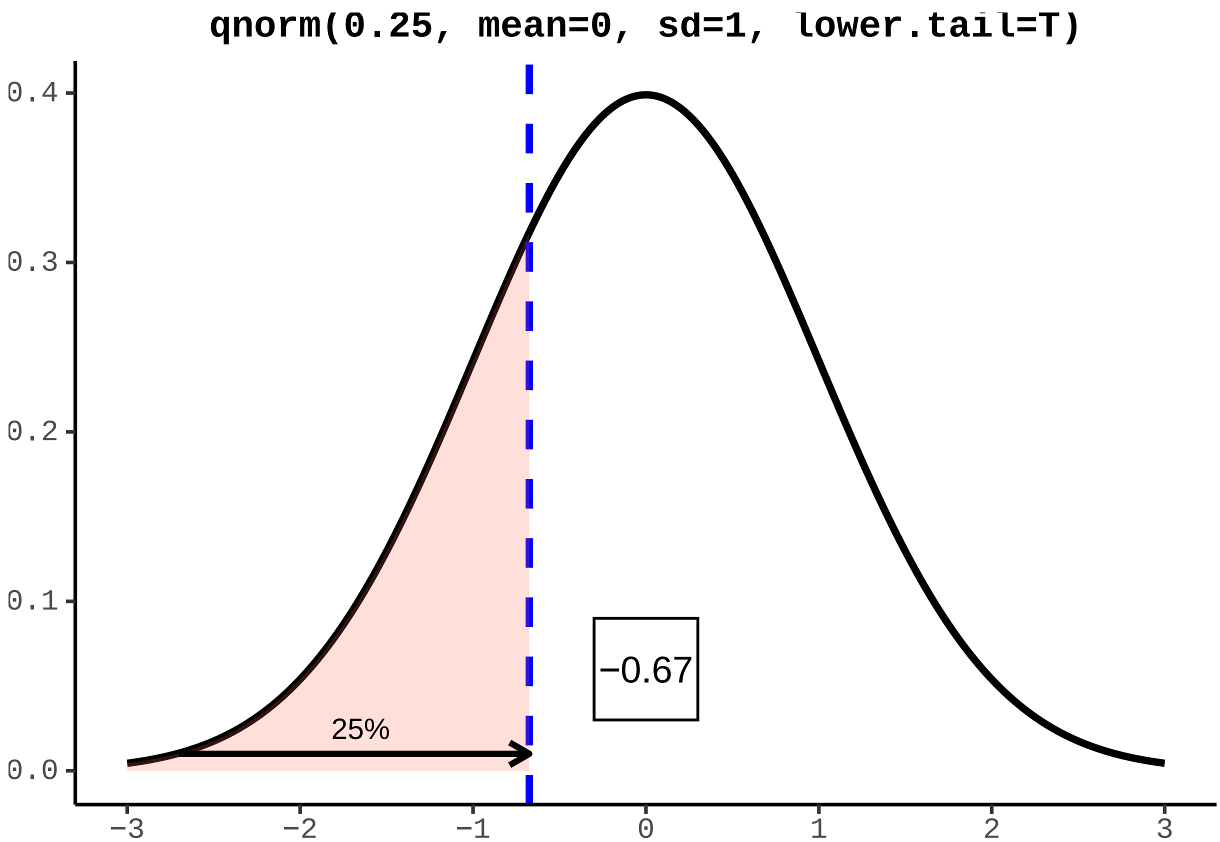
As per usual, the default values for the arguments `mean` and `sd` are 0 and 1, respectively (thus drawing the probability distribution from a standard normal distribution). So, if I want the quantile value corresponding to the cumulative probabilty of `0.5`, I will have to say:

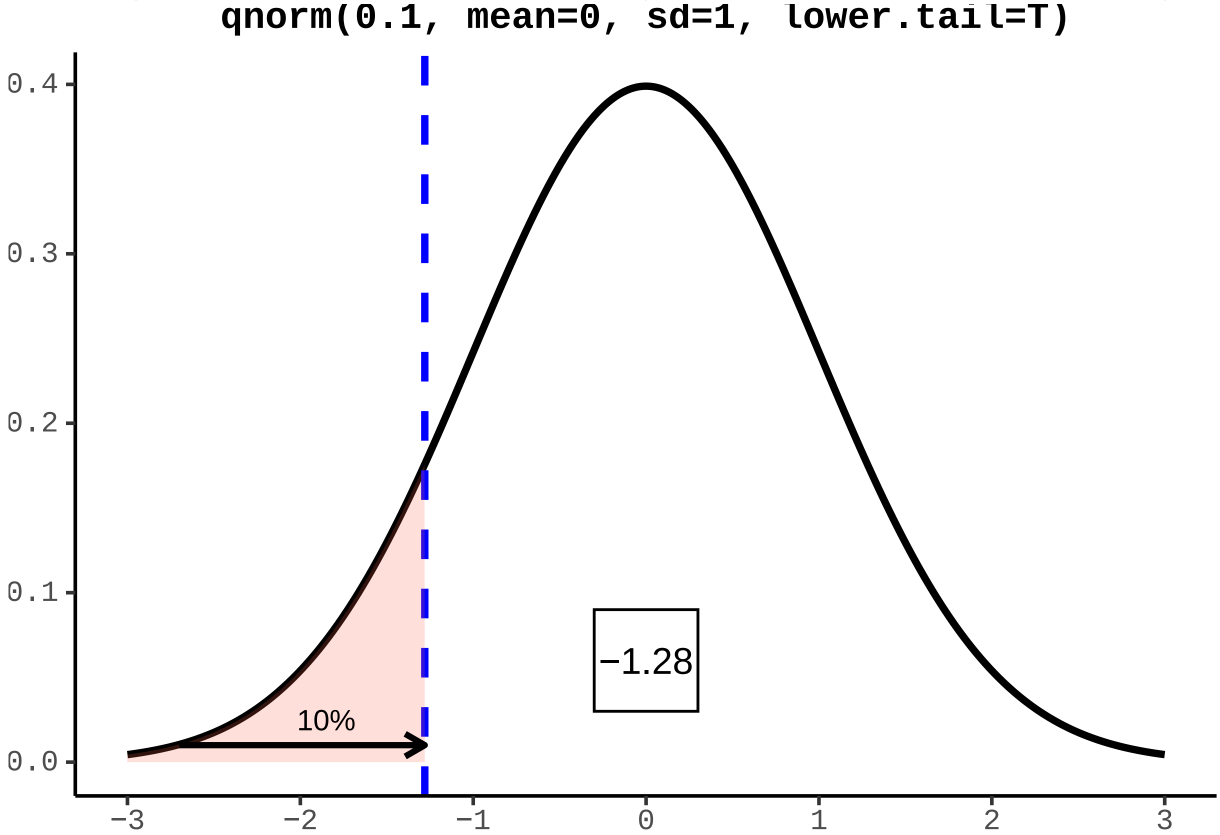
```
qnorm(p=0.5, mean=0, sd=1, lower.tail=T)
```

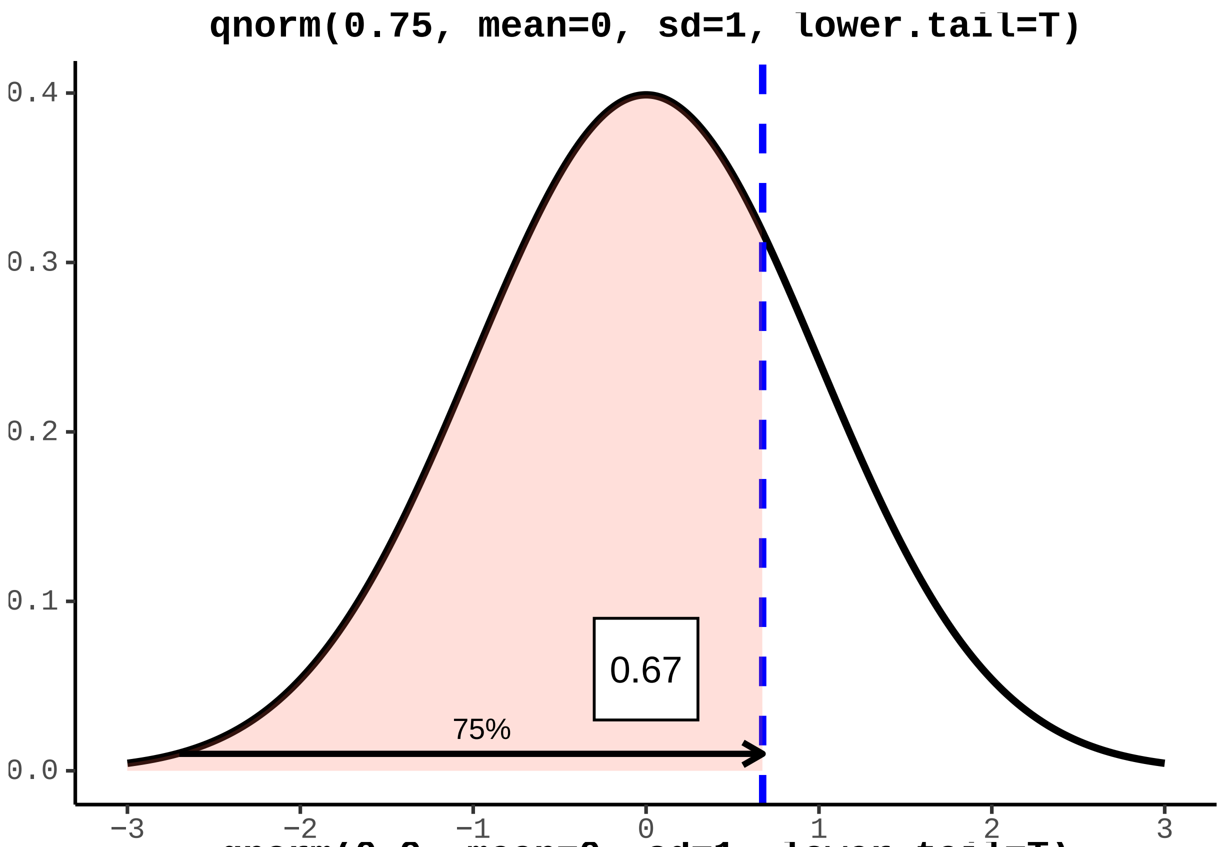
```
## [1] 0
```



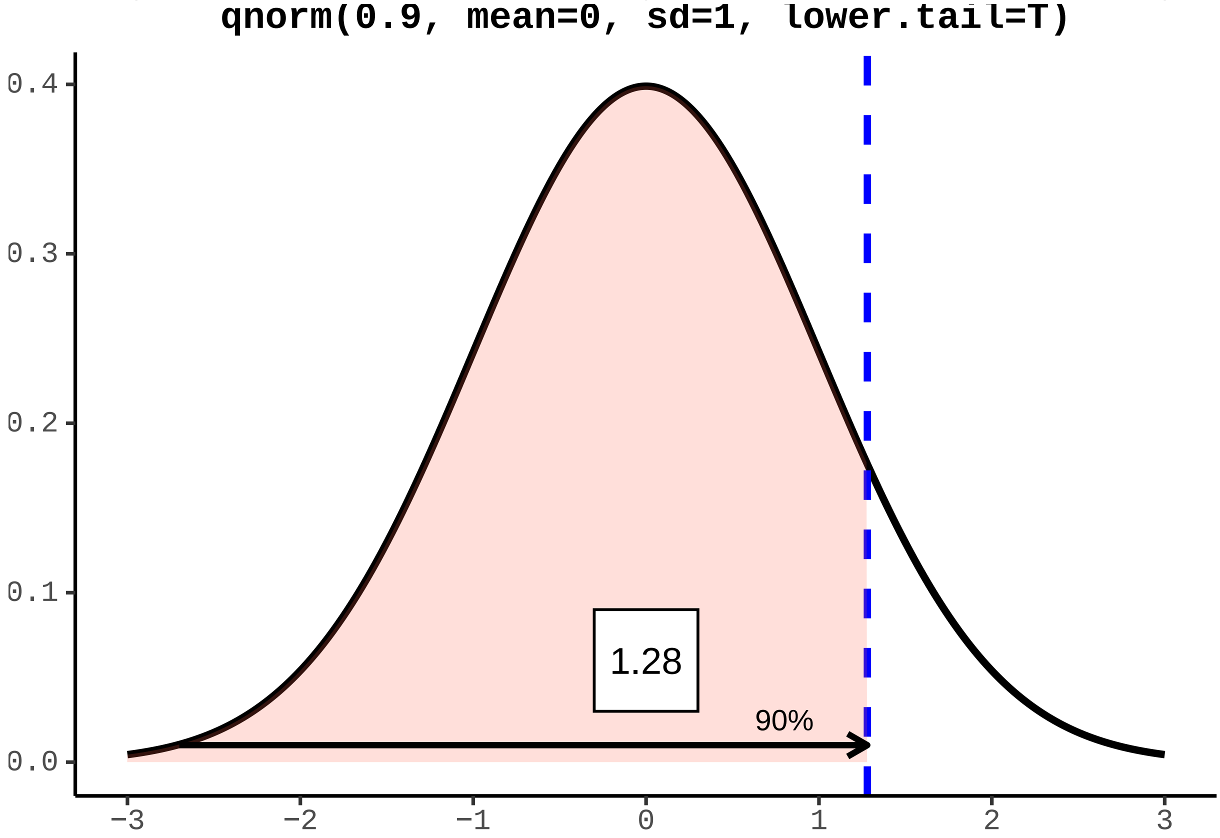
Here, the same thing we talked about above regarding the `lower.tail` argument applies here too. `lower.tail=TRUE` will give you the quantile corresponding to the lower/leftward tail; `lower.tail=FALSE` will give you the quantile corresponding to the upper/rightward tail.


qnorm(0.25, mean=0, sd=1, lower.tail=T)


qnorm(0.1, mean=0, sd=1, lower.tail=T)



qnorm(0.75, mean=0, sd=1, lower.tail=T)



qnorm(0.9, mean=0, sd=1, lower.tail=T)

#Other distributions

While the plots above are specific to the standard normal distribution, the same concepts apply to distribution. In this class in particular, we will be dealing with two other distributions:

1. the t-distribution: `rt()`, `pt()`, `qt()`, `dt()`
2. the F-distribution: `rf()`, `pf()`, `qf()`, `df()`

But there are many more: the beta distribution, the binomial distribution, the gamma distribution, the logistic distribution, the Poisson distribution. While you are not asked to remember these names, it will be useful to remember that, no matter what kind of distribution you will end up dealing with in the case, in R they all have the same functions (called the *pqrd family*)!