# Contents

# 1 Background

Before explaining quantile plots, some background information is in order. You may skip to any section you need, but the main focus will be on cumulative distribution functions and quantile functions.

## 1.1 Random variables

Random variables are variables that could take on a range of values, until they are 'realised'. We typically represent them with capital letters, e.g. $X_1, \ldots, X_n$ are instances of random variables.

The realisations of these random variables are typically denoted with lowercase letters. Referencing the above, when $X_1$ is 'realised' and takes on a value, we denote it with $x_1$.

## 1.2 Distributions

Distributions describe what values random variables can take on, and with what probability.

A simple example: a random variable can take on one of two values: $0$ or $1$, with equal probability ($p = 0.5$). We describe variables that have these exact specifications as *Bernoulli distributed* with parameter $p = 0.5$.

If a random variable $X_1$ follows this distribution, we notate it as $X_1 \sim \text{Bernoulli}(0.5)$. There are many other distributions, some of which describe physical phenomena. Some are discrete, i.e. they take on discrete values (e.g. $1, 2, 3, \ldots$), and others are continuous, i.e. any decimal value within a range (e.g. any number in the interval $(0, 1)$, or any real number).

Examples of discrete distributions:

- Binomial distribution
- Geometric distribution

Examples of continuous distributions:

- Normal distribution
- Uniform distribution
- Exponential distribution
- Gamma distribution

## 1.3 Probability density functions (p.d.f.)

Every distribution has a probability density functions, which expresses the probability it can take on a value. Following the example of the Bernoulli distribution, say $X_1 \sim \text{Bernoulli}(p)$, the p.d.f. can be expressed as

$$f(x) = p^x(1-p)^{1-x},\ x \in \{0, 1\}$$

Fixing a value of $p$, say $0.4$, will yield to us that

$$f(x) = 0.4^x(1-0.4)^{1-x} \implies f(0) = 0.4 \text{ and } f(1) = 0.6$$

i.e. that $X_1$ has a 40% probability of taking on the value $0$ (represented by $f(0) = 0.4$) and 60% probability of taking on the value $1$ (represented by $f(1) = 0.6$).

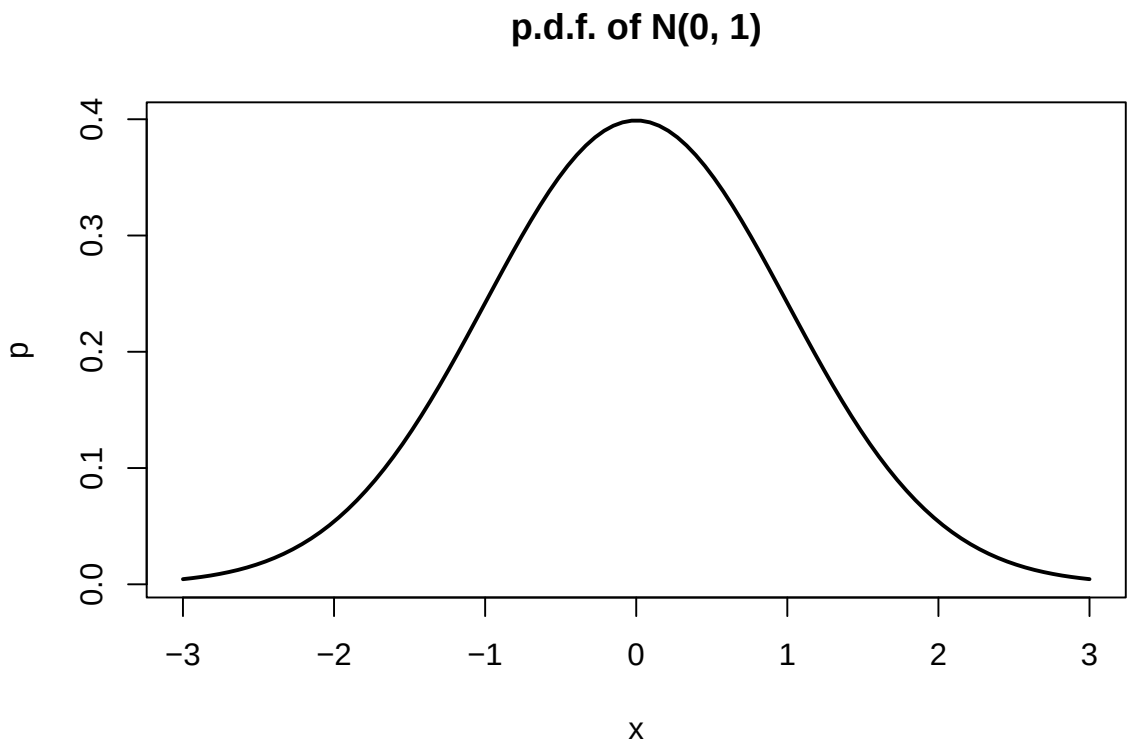
Typically it is denoted with a lowercase $f$ and an optional subscript, e.g. $f_X(x)$ to represent the p.d.f. of $X$.

For discrete distributions, this is also called the probability mass function (p.m.f.), but they represent the same idea.

One important property of a p.d.f. or p.m.f. is that the sum of all probabilities must equal to 1.

A brief example of the p.d.f. of the standard normal:

```
x <- seq(-3, 3, length.out = 100)
p <- dnorm(x)
plot(x, p, type='l', lwd=2, main="p.d.f. of N(0, 1)")
```




### 1.4 Cumulative distribution functions (c.d.f.)

The cumulative distribution is just the summation (for discrete distributions) or integral (for continuous distributions) of all probabilities up until some number. With reference to a p.d.f. $f_X$, the c.d.f. $F_X$ is defined as:

$$F_X(x) = \int_{-\infty}^{x} f_x(t)dt$$

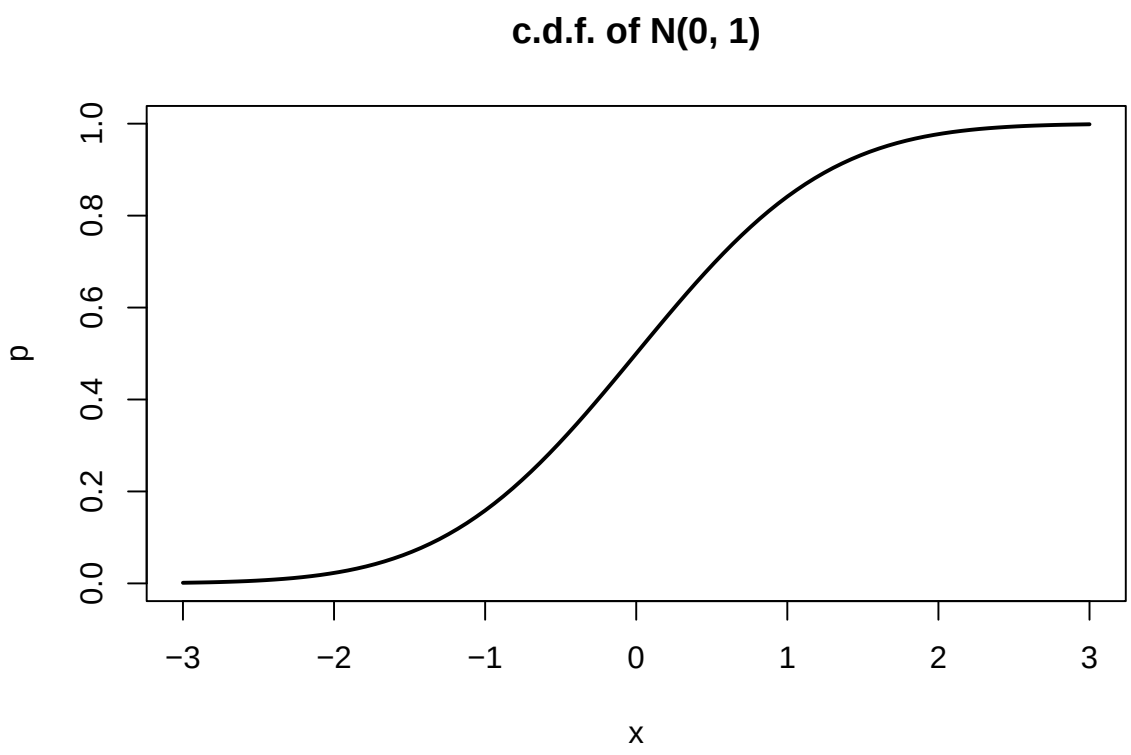
This represents **exactly** the area under the curve of the p.d.f., and so we must have that a c.d.f. approaches 1 as $x \to \infty$, i.e.

$$\lim_{x\to\infty} F_X(x) = 1$$

Note that this is an increasing function, since we can never have negative probabilities.

A brief example of the p.d.f. of the standard normal:

```r
x <- seq(-3, 3, length.out = 100)
p <- pnorm(x)
plot(x, p, type='l', lwd=2, main="c.d.f. of N(0, 1)")
```




### 1.5 Quantile functions

Quantile functions, simply put, are inverses to c.d.f. functions. Take for example, the standard normal distribution c.d.f. It takes in a value within its state space, and outputs a probability.

$$F : \mathbb{R} \to [0,1] : F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$$

The quantile function is thus the inverse of this, i.e. takes in a probability, and outputs a value in its state space. I don't really know common notation for this, so I'll just call it $F^{-1}$:

$$F^{-1} : [0, 1] \to \mathbb{R}$$

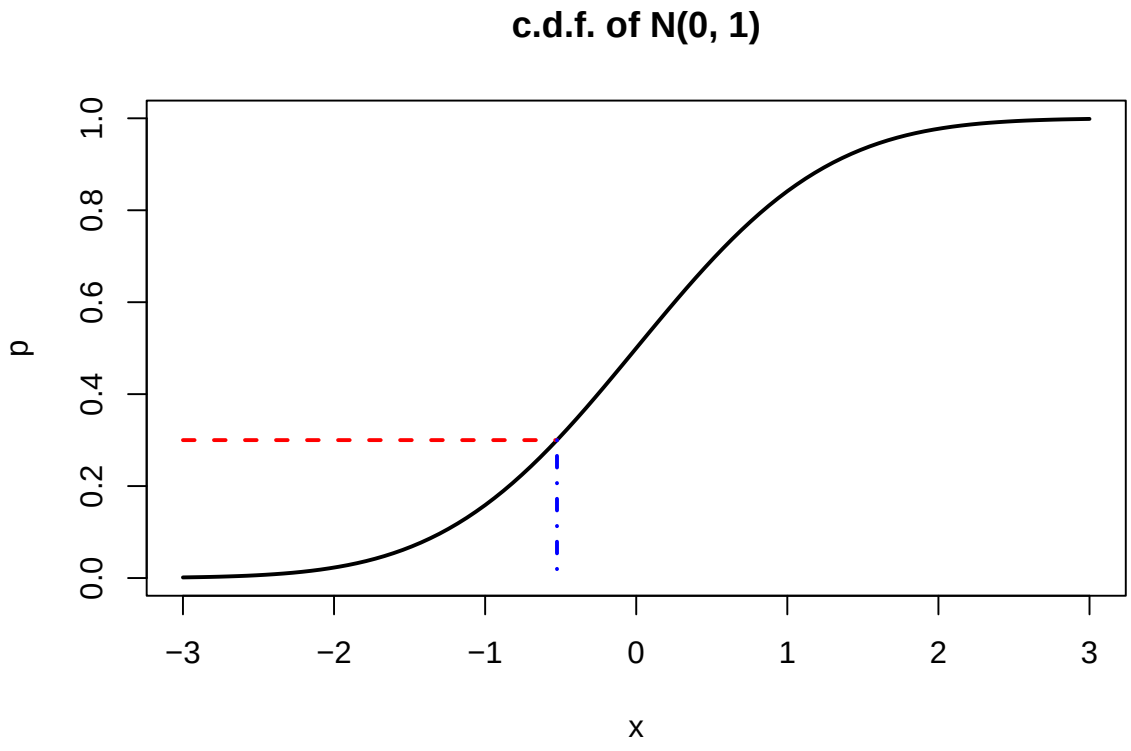
The closed form of this quantile function for the normal distribution involves cursed math that is not in the interests of this course.

Visualising the procedure of getting the inverse is simple: take the c.d.f. and draw horizontal and vertical lines. For example, I want the 0.3 quantile of the standard normal, i.e. some $x$ such that $F(x) = 0.3$ or equivalently $x = F^{-1}(0.3)$.

We draw the horizontal line at $p = 0.3$ (the red dashed line below) and find where it intersects the c.d.f. A horizontal line is drawn where it intersects the c.d.f. (the blue dotdashed line below), which corresponds to the $x$ we desired such that $F(x) = 0.3$.

```r
x <- seq(-3, 3, length.out = 100)
p <- pnorm(x)
plot(x, p, type='l', lwd=2, main="c.d.f. of N(0, 1)")

q <- 0.3
x.q <- qnorm(0.3)
segments(-3, q, x.q, q, lty="dashed", col="red", lwd=2)
segments(x.q, q, x.q, 0, lty="dotdash", col="blue", lwd=2)
```



### 1.6 Distributions in R

R has information on many distributions. Focusing on the normal distribution, we have information on:

- the p.d.f. with `dnorm()`;
- the c.d.f. with `pnorm()`;
- the quantile function (inverse c.d.f.) with `qnorm()`; and
- random number generation with `rnorm()`.

All other distributions in R follow the same prefix style with d, p, q, and `r`. You can use `?distributions` to see all distributions supported in R.

## 2 Quantile-quantile plots

It might be helpful to understand how quantile-quantile plots (qqplots) are created, and their purpose. We will avoid using qq-functions in this section.

Generally, the purpose of a qqplot is to compare the distribution of two variables, *usually* with reference to a theoretical distribution. When a theoretical distribution is not available, one may rely on an empirical distribution (based on data).

### 2.1 Testing normality of data

Say we have some data, and wish to see whether it follows a normal distribution. We may look at a histogram or a density plot and see if it looks normal, but let's use quantile-quantile plots instead.

The idea is to have your data plotted against a reference distribution. In this case, we know a closed form of the distribution's p.d.f. and c.d.f., so we will use those instead.

```
x.data <- c(28,29,30,29,30,30,26,28,29,29,24,28,29,27,30,29,28,
            26,30,28,28,30,24,27,29,26,27,30,31,27,31,30,26,26)
```

### 2.2 Sample quantiles

The first step is to get probabilities that are spaced out nicely. A rough idea on why the probabilities should be spaced out nicely:

For any dataset, we can compute (sample) quantiles. For example, for the above, we can use the `quantile()` function to compute different quantiles:

```
quantile(x.data, ppoints(length(x.data)))
```

```
## 1.470588% 4.411765% 7.352941% 10.29412% 13.23529% 16.17647% 19.11765% 22.0588
##  24.00000  24.91176  26.00000  26.00000  26.00000  26.00000  26.30882  27.000
##       25% 27.94118% 30.88235% 33.82353% 36.76471% 39.70588% 42.64706% 45.5882
##  27.00000  27.00000  27.19118  28.00000  28.00000  28.00000  28.00000  28.000
## 48.52941% 51.47059% 54.41176% 57.35294% 60.29412% 63.23529% 66.17647% 69.1176
##  28.01471  28.98529  29.00000  29.00000  29.00000  29.00000  29.00000  29.000
## 72.05882%       75% 77.94118% 80.88235% 83.82353% 86.76471% 89.70588% 92.6470
##  29.77941  30.00000  30.00000  30.00000  30.00000  30.00000  30.00000  30.000
## 95.58824% 98.52941%
##  30.54412  31.00000
```

These quantiles are dependent on the data, and nothing else. If the data were normally distributed, then the quantiles should be (roughly) matching, because if the quantiles match, then the data follow a normal distribution.

We use `ppoints()` here to split the space up nicely, so that we avoid $p = 0$ and $p = 1$ - in many distributions, the quantile function $F^{-1}$ usually has $F^{-1}(0) = -\infty$ and $F^{-1}(1) = +\infty$, with the normal distribution being such an example.

Since we have 34 points of data, we also want 34 nicely spaced quantiles.

```r
q <- ppoints(length(x.data))
round(q, 4)
```

```
##  [1] 0.0147 0.0441 0.0735 0.1029 0.1324 0.1618 0.1912 0.2206 0.2500 0.2794
## [11] 0.3088 0.3382 0.3676 0.3971 0.4265 0.4559 0.4853 0.5147 0.5441 0.5735
## [21] 0.6029 0.6324 0.6618 0.6912 0.7206 0.7500 0.7794 0.8088 0.8382 0.8676
## [31] 0.8971 0.9265 0.9559 0.9853
```

We have nicely separated quantiles, and now we need points that correspond to the quantiles of the reference distribution. Again, we want to compare it to the normal distribution, so we use the quantiles of the normal distribution:

```r
q.x <- qnorm(q)
round(q.x, 4)
```

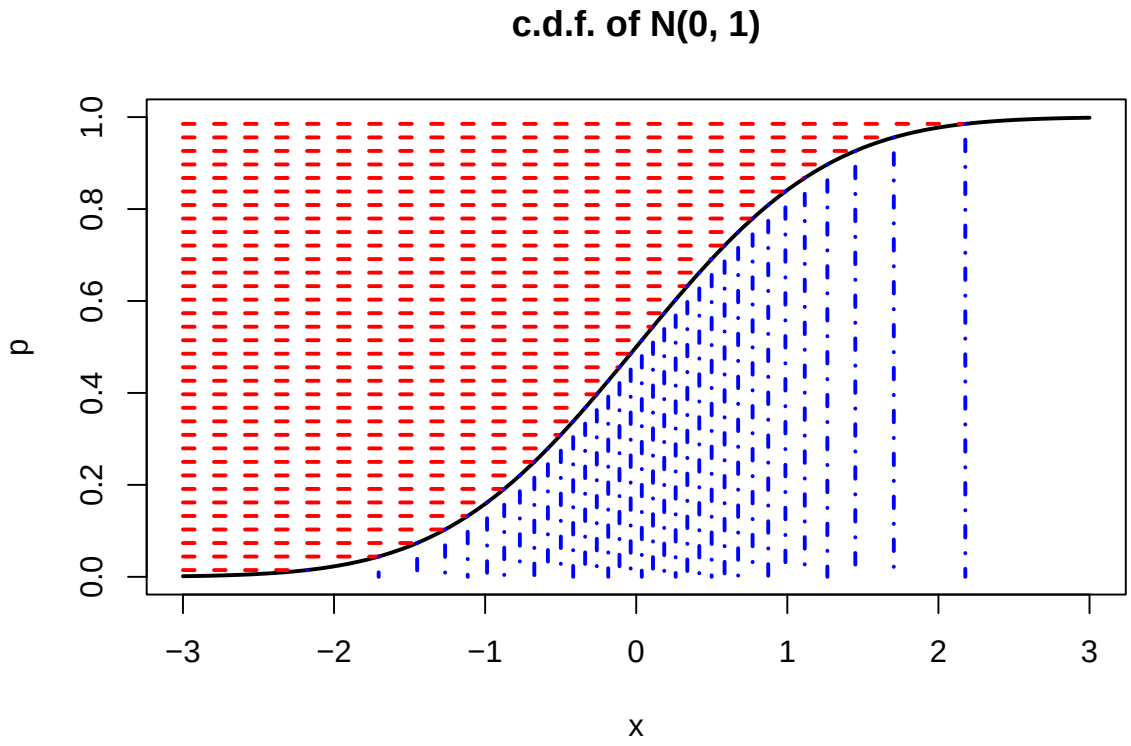
```
##  [1] -2.1779 -1.7048 -1.4500 -1.2650 -1.1153 -0.9872 -0.8736 -
0.7702 -0.6745
## [10] -0.5846 -0.4992 -0.4173 -0.3381 -0.2610 -0.1854 -0.1108 -
0.0369  0.0369
## [19]  0.1108  0.1854  0.2610  0.3381  0.4173  0.4992  0.5846  0.6745  0.7702
## [28]  0.8736  0.9872  1.1153  1.2650  1.4500  1.7048  2.1779
```

Each value in q now represents the value at which the c.d.f. was equal to the probabilities in p. For example, looking at the first values of p and q, we have $F(-2.1779) = 0.0147$ or equivalently $-2.1779 = F^{-1}(0.0147)$.

Visually represented, we took all the quantiles and mapped them to the corresponding x values.

```
x <- seq(-3, 3, length.out = 100)
p <- pnorm(x)
plot(x, p, type='l', lwd=2, main="c.d.f. of N(0, 1)")

segments(-3, q, q.x, q, lty="dashed", col="red", lwd=2)
segments(q.x, q, q.x, 0, lty="dotdash", col="blue", lwd=2)
```




Now we have the theoretical quantiles, now we want to compare it to the sample quantiles. And instead of the sample quantiles above, we can just use our data directly - all we need to do is *arrange* it in ascending order.

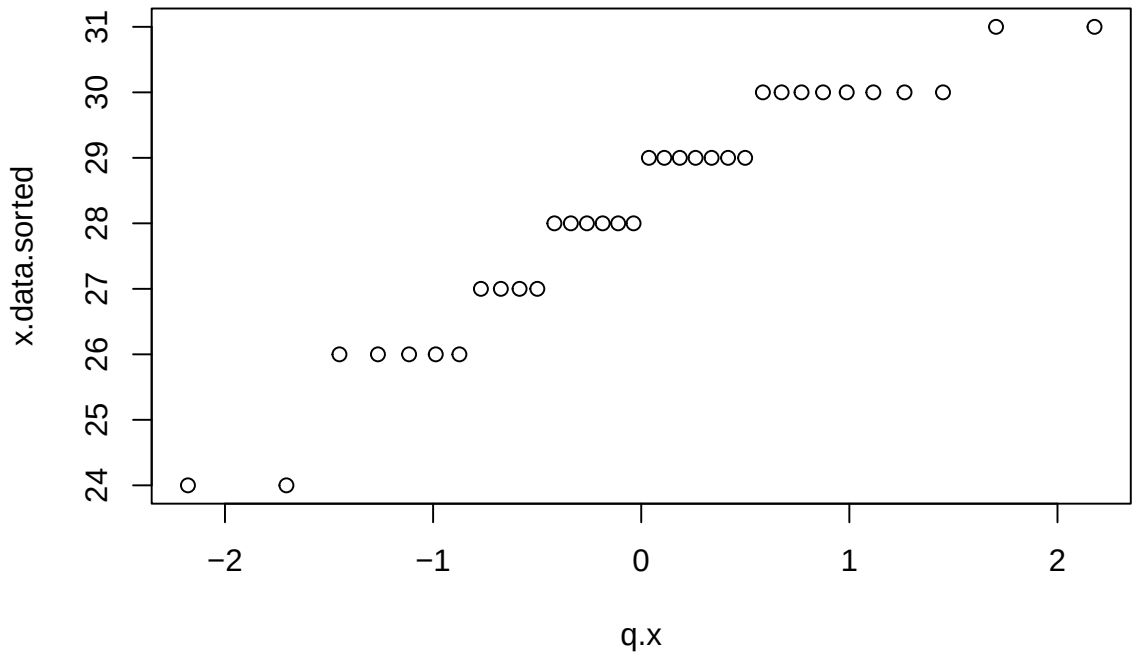
Each point plotted on a quantile-quantile plot represents the same quantiles of the two distributions - the sample data and the theoretical or reference distribution.

```
x.data.sorted <- sort(x.data)
# We are essentially plotting this dataframe.
data.frame(quantile=q, q.x, x.data.sorted)
```

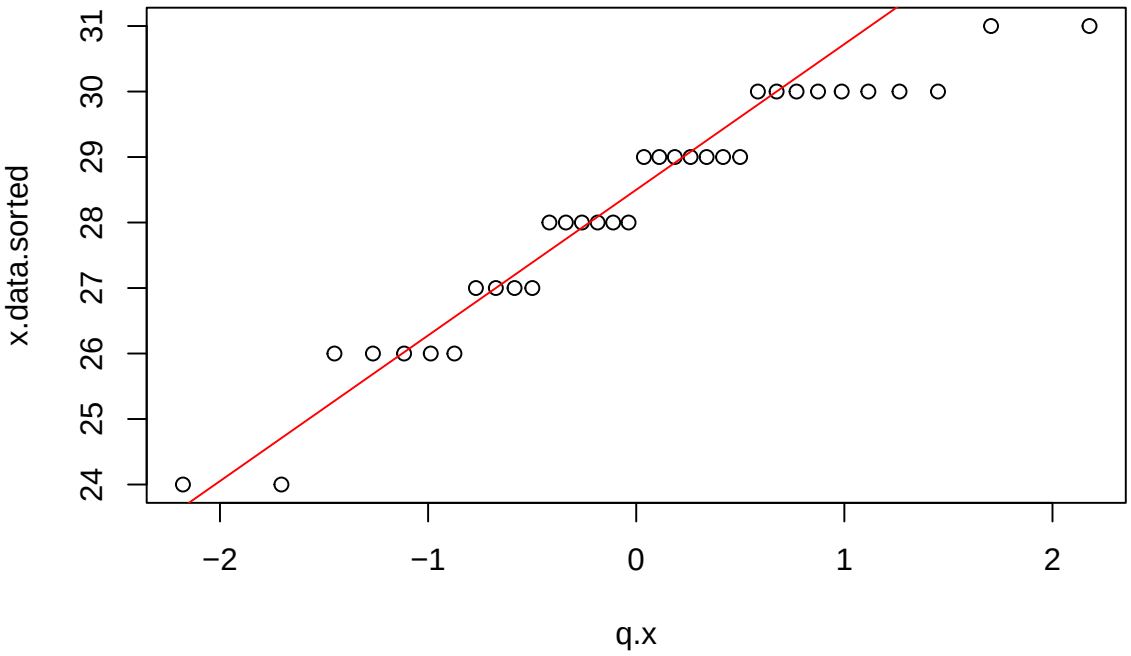
```
## # A tibble: 34 x 3
```

```
##    quantile    q.x x.data.sorted
##       <dbl>  <dbl>         <dbl>
##  1   0.0147 -2.18             24
##  2   0.0441 -1.70             24
##  3   0.0735 -1.45             26
##  4   0.103  -1.26             26
##  5   0.132  -1.12             26
##  6   0.162  -0.987            26
##  7   0.191  -0.874            26
##  8   0.221  -0.770            27
##  9   0.25   -0.674            27
## 10   0.279  -0.585            27
## # i 24 more rows
```

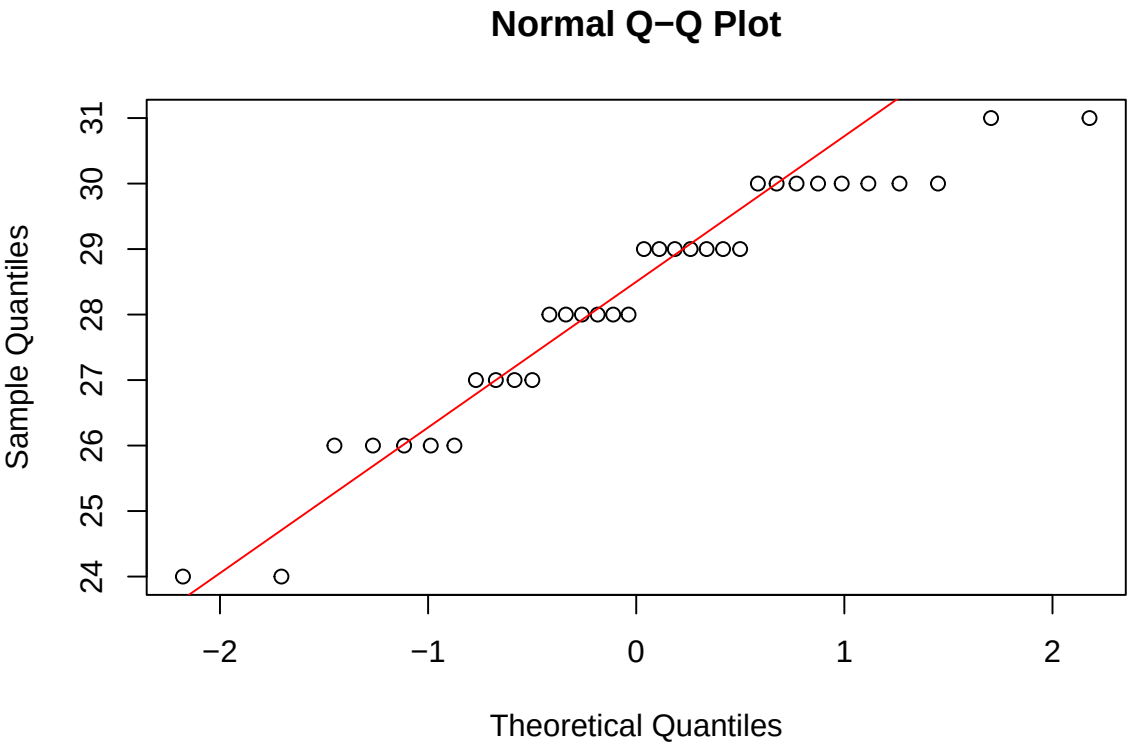
```r
plot(q.x, x.data.sorted)
```




```r
# The below produce *exactly the same graphs*.
qqplot(q.x, x.data.sorted)
qqline(x.data.sorted, col="red")
```




```r
qqnorm(x.data.sorted)
qqline(x.data.sorted, col="red")
```



Ultimately, each point plotted in the quantile-quantile plots represent a pair of values, sorted in order, and represent the same quantiles, of the respective distributions.

### 2.3 qqnorm()

`qqnorm()` is a simple shorthand process for exactly what we did above. It uses the normal distribution as the reference distribution.

### 2.4 qqline()

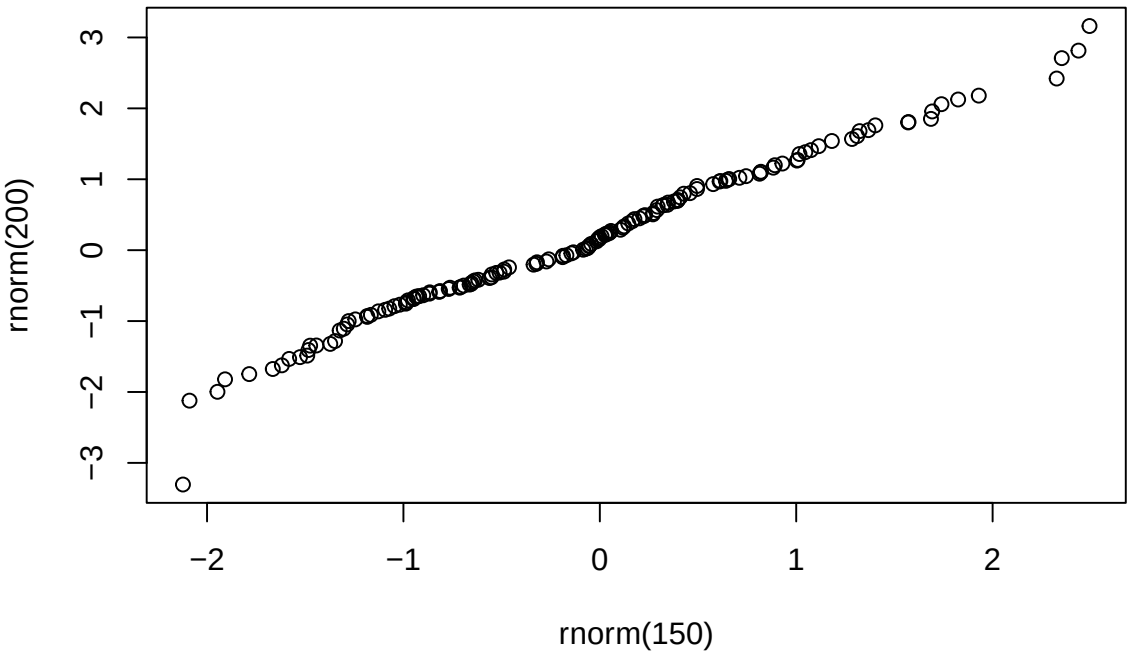
`qqline()` just draws a line between the 25th and 75th quantiles. That's really it. However, the reference distribution is required to get the other coordinate: on one axis, we have the theoretical quantiles, on the other, the sample quantiles.

So we have one piece of information (the sample quantiles), and need to generate the other (the theoretical quantiles). By default, the normal quantiles are used, but can be changed by changing the `distribution` parameter (e.g. `qunif`, `qchisq`, `qexp`).

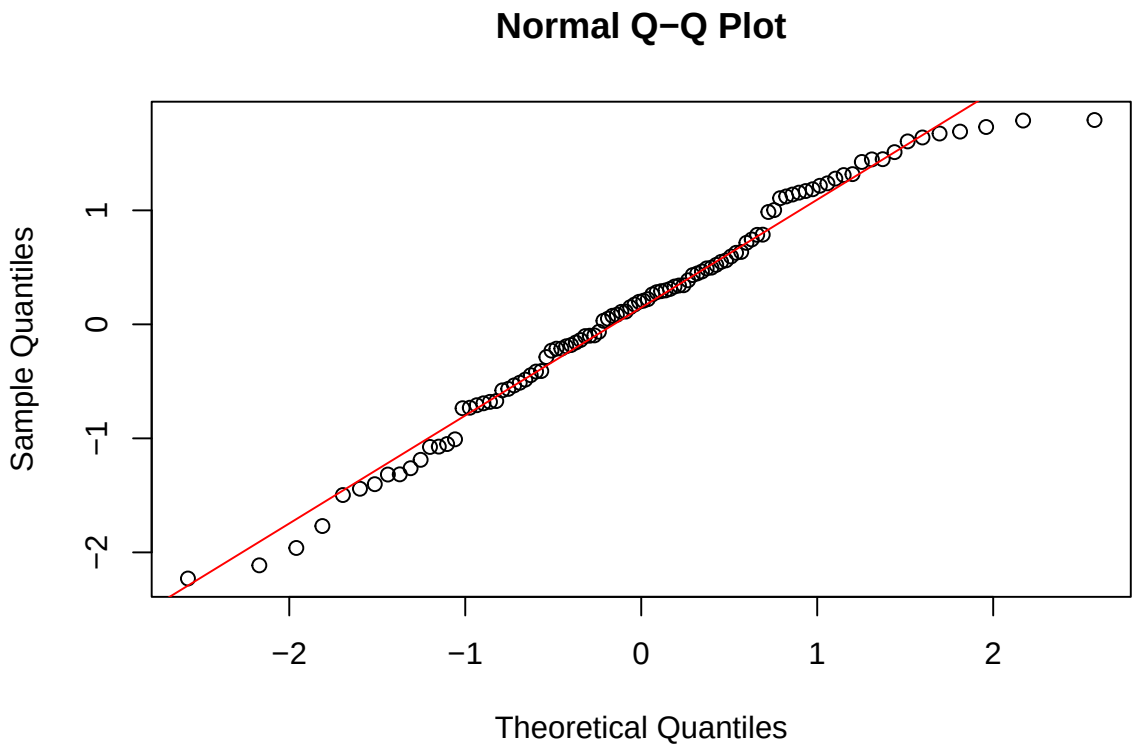
### 2.5 qqplot()

`qqplot()` requires you to pass two sets of data, and is used in the case where the reference distribution is another set of data. In general, it just orders the data, and pairs it together. I'm not really sure what happens when one dataset is larger than the other.

```r
qqplot(rnorm(150), rnorm(200))
```




```r
x <- rnorm(100)
qqnorm(x)
qqline(x, col="red")
```




Once you know how the quantile-quantile plot is created, interpreting it should come more naturally.

### 2.6 Interpretation

The quantile-quantile plot gives us a quick view of how a set of values is distributed with reference to another distribution.

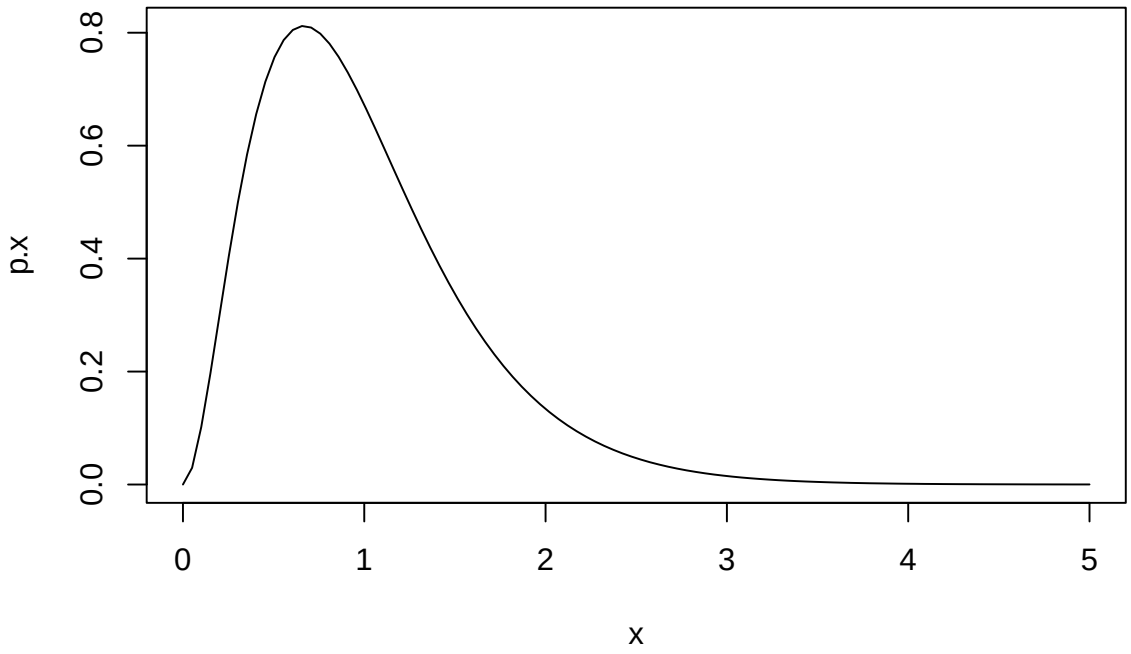
If the two distributions are similar, then the plot should have a nice line forming.

## 3 Shorter/longer tails

Shorter and longer tails refer to the skewness of a distribution. For example, the normal distribution is symmetric about its mean, and so does not have a 'long' or 'short' tail to speak of.

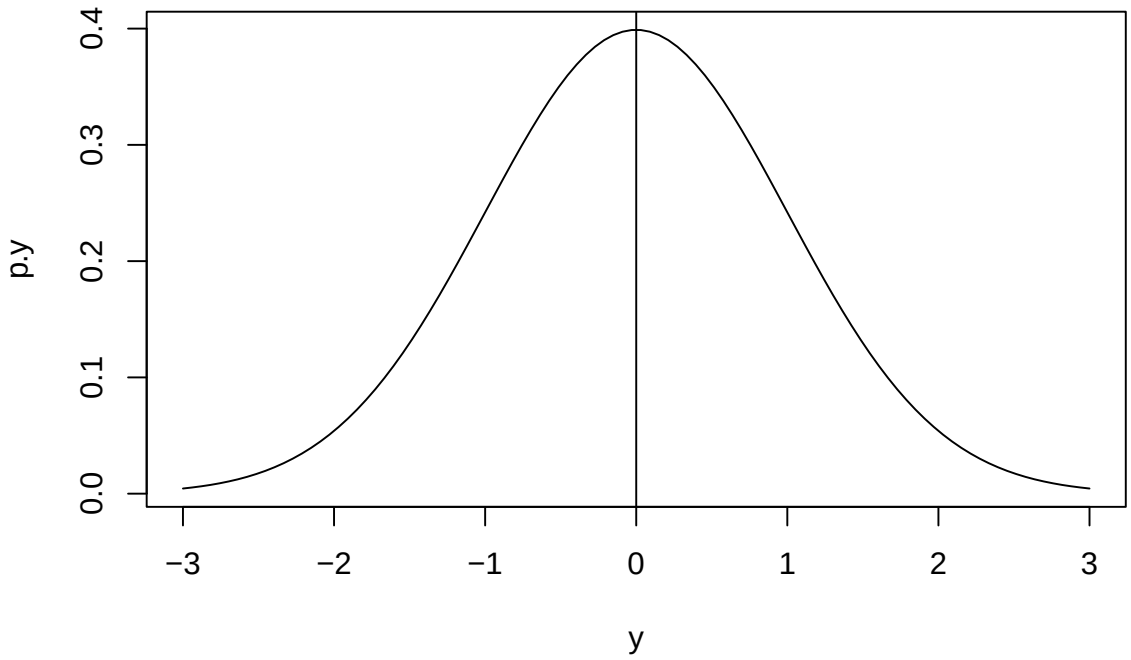
On the other hand, the gamma distribution is not symmetric. The lowest value it can have is 0, but the upper bound is theoretically infinite, and thus the gamma distribution has a *long* right tail.

```r
x <- seq(0, 5, length.out = 100)
p.x <- dgamma(x, 3, rate=3)
plot(x, p.x, type='l')
```

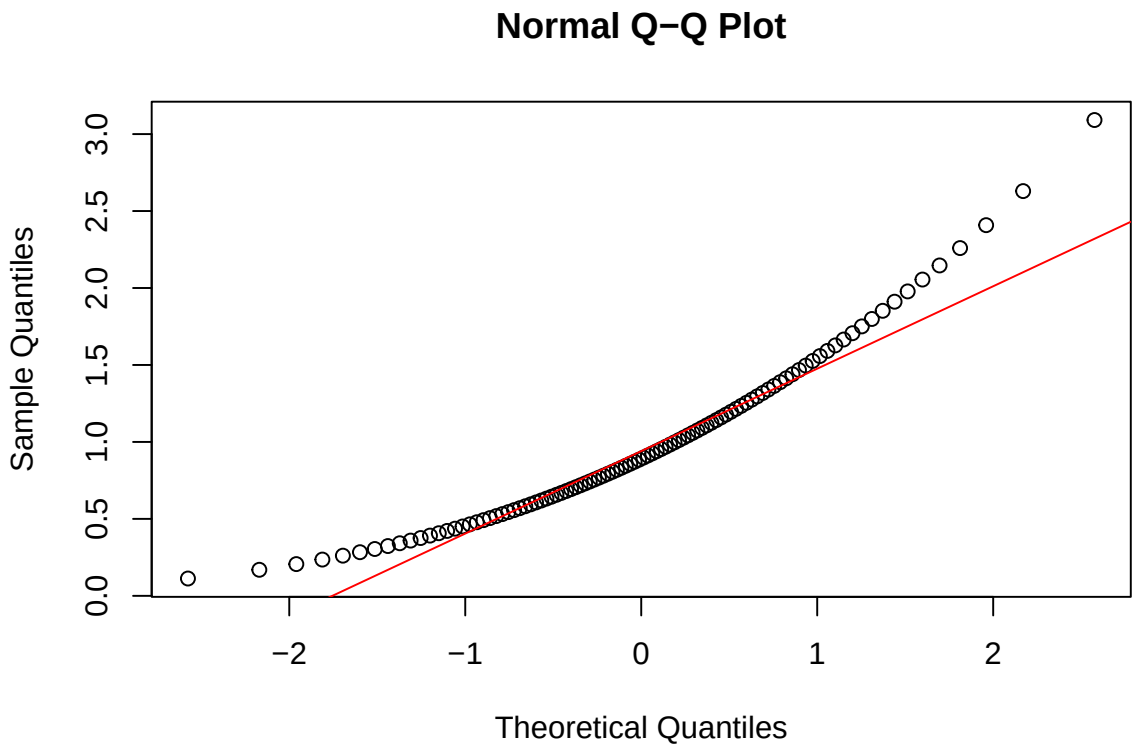



```r
# abline(v=1)

y <- seq(-3, 3, length.out = 100)
p.y <- dnorm(y)
plot(y, p.y, type='l')
abline(v=0)
```




```r
q.x <- qgamma(ppoints(100), 3, rate=3)
qqnorm(q.x)
qqline(q.x, col="red")
```


Normal Q−Q Plot
Sample Quantiles
0.0
0.5
1.0
1.5
2.0
2.5
3.0
−2
−1
0
1
2
Theoretical Quantiles