# STAT 210
# Applied Statistics and Data Analysis:
# Week 3 - Summary

Joaquín Ortega

joaquin.ortegasanchez@kaust.edu.sa

# Announcements

Tutorials

Zoom link in Blackboard

- Saturday, September 17, Tutorial on `R`
- Saturday, September 24, Tutorial on graphs in `R`

# Graphs in R

```
plot
```

`plot()` is the standard function for plotting in `R`.

What you get depends on the object that holds the data, the mode of the data, and the syntax you use.

```
plot(Sepal.Length, Sepal.Width)
plot(Sepal.Width ~ Sepal.Length)
plot(~ Sepal.Length + Sepal.Width)
```

```
plot(Species) # Barplot
plot(Species, Sepal.Length) # Boxplot
plot(iris) # Matrix of plots
plot(Petal.Length ~ Sepal.Width + Sepal.Length) # Two plots
```

The `type` option determines the type of plot to be produced. The options are listed in the table.

| Option | Value |
|---|---|
| `type = 'p'` | Plots points, is the default option |
| `type = 'l'` | Plots lines. |
| `type = 'b'` | Plots points joined by lines. |
| `type = 'o'` | Points and lines are superimposed. |
| `type = 'h'` | Vertical lines. |
| `type = 's'` | Step function, continuous from right. |
| `type = 'S'` | Step function, continuous from left. |
| `type = 'n'` | Does not draw the graph but keeps the dimensions |

# Options in `plot`

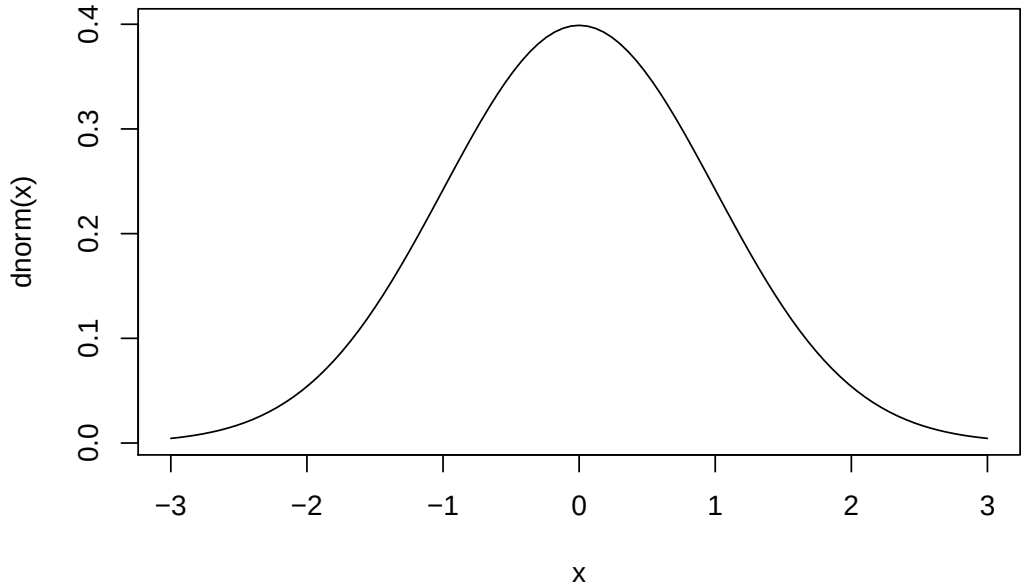
- `xlab` and `ylab`
- `main` and `sub`
- `xlim` and `ylim`
- `asp`
- `lty` and `lwd`
- `pch`
- `col`

# High-level commands

## curve

```
curve(expr, from, to, add = FALSE, ...)
```

```
curve(dnorm(x), -3, 3)
```



## Boxplots

```
boxplot(formula, data = NULL, ...)
```

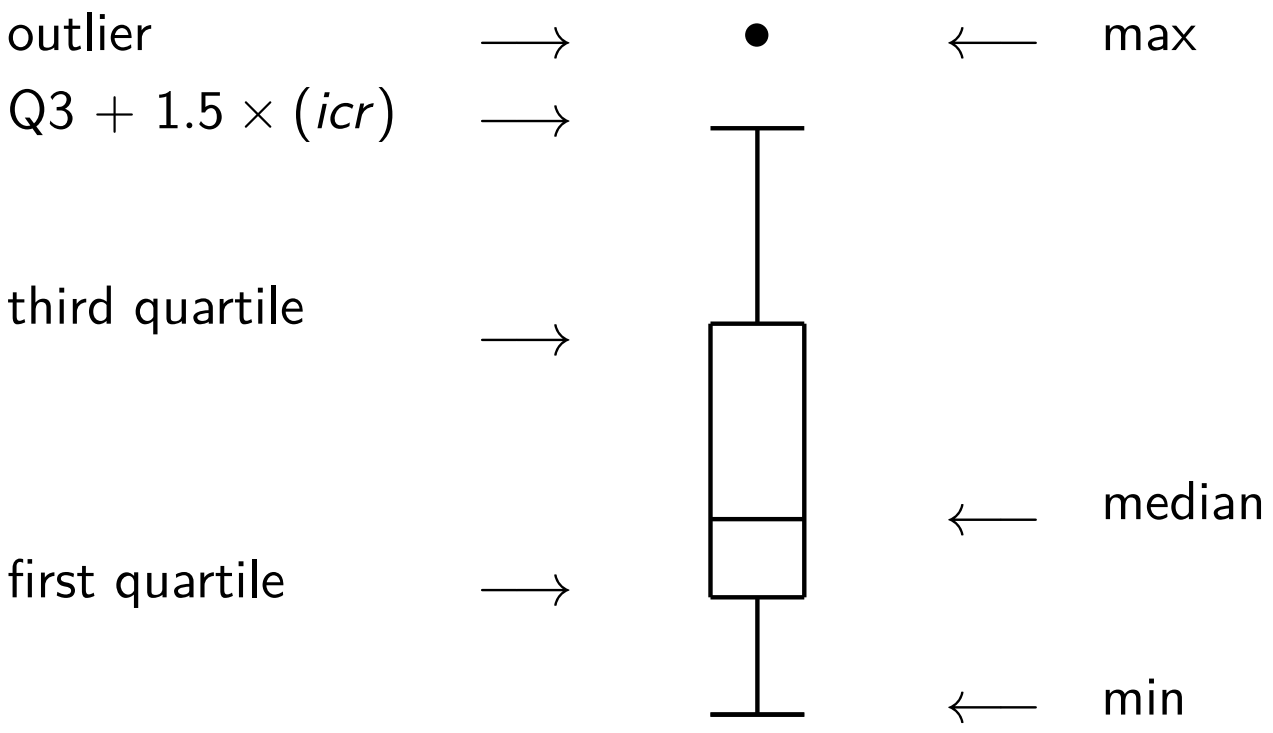



Figure 4.1

barplot

```
barplot(height, ...)
```

hist (See also truehist in MASS)

```
hist(x, breaks = "Sturges", freq = NULL,
     probability = !freq, ...)
```

dotchart

```
dotchart(x, labels = NULL, groups = NULL, ...)
```

pairs

```
pairs(formula, data = NULL, ..., subset,
      na.action = stats::na.pass)
```

## contour

```
contour(x = seq(0, 1, length.out = nrow(z)),
        y = seq(0, 1, length.out = ncol(z)),
        z, nlevels = 10)
```

## filled.contour

```
filled.contour(x = seq(0, 1, length.out = nrow(z)),
        y = seq(0, 1, length.out = ncol(z)),
        z, nlevels = 10,)
```

## persp

```
persp(x = seq(0, 1, length.out = nrow(z)),
      y = seq(0, 1, length.out = ncol(z)),
      z, xlim = range(x), ylim = range(y),
      zlim = range(z, na.rm = TRUE)...)
```

```
sunflowerplot(x,y)
```

```
stripchart(x)
```

```
matplot(x,y)
```

```
plot.ts(x)
```

```
image(x,y,z)
```

```
stars(x)
```

# Low-level commands

legend

```
legend(x, y = NULL, legend, fill, col, bg)
```

points

```
points(x, y = NULL, type = 'p', ...)
```

lines

```
lines(x, y = NULL, type = "l", ...)
```

abline

```
abline(a = NULL, b = NULL, h = NULL, v = NULL,
       reg = NULL, coef = NULL,  ...)
```

# Other Commands

axis()

```
axis(side, at = NULL, labels = TRUE, tick = TRUE,...)
```

text()

```
text(x, y = NULL, labels = seq_along(x$x),
     adj = NULL, ...)
```

title

```
title(main = NULL, sub = NULL, xlab = NULL,
      ylab = NULL, line = NA, outer = FALSE, ...)
```

arrows

```
arrows(x0, y0, x1 = x0, y1 = y0, length = 0.25,
       angle = 30, code = 2, col = par("fg"),
       lty = par("lty"),...)
```

# Graphical Parameters

```
par()
```

```
old.par <- par(no.readonly = TRUE)
plot(cars)
par(bg=7, bty='u', cex=1.5, col='blue', col.axis=4,
    font=2, lty='dashed', lwd=3, pch=3, las=2, tck=1)
plot(cars)
par(old.par)
plot(cars)
```

# Graphical Windows

`mfrow` or `mfcol` in `par`.

```
par(mfrow(c(m,n)))
par(mfcol(c(m,n)))
```

`split.screen`

```
split.screen(c(m,n)
```

`layout`

```
layout(mat, ...)
```

# Interactive functions

## locator

```
locator(n = 512, type = "n", ...)
```

## identify

```
identify(x, ...)
```

# Quantile plots

# Location and Scale Family

Given a random variable $X$ with distribution function $F$, the location and scale family associated to $F$ is the family of distributions of the variables

$$aX + b$$

where $a \neq 0$ and $b \in \mathbb{R}$.

We say that $b$ is a **location** parameter while $a$ is a **scale** parameter.

Example:

The family of normal distributions $N(\mu, \sigma^2)$.

# Quantiles

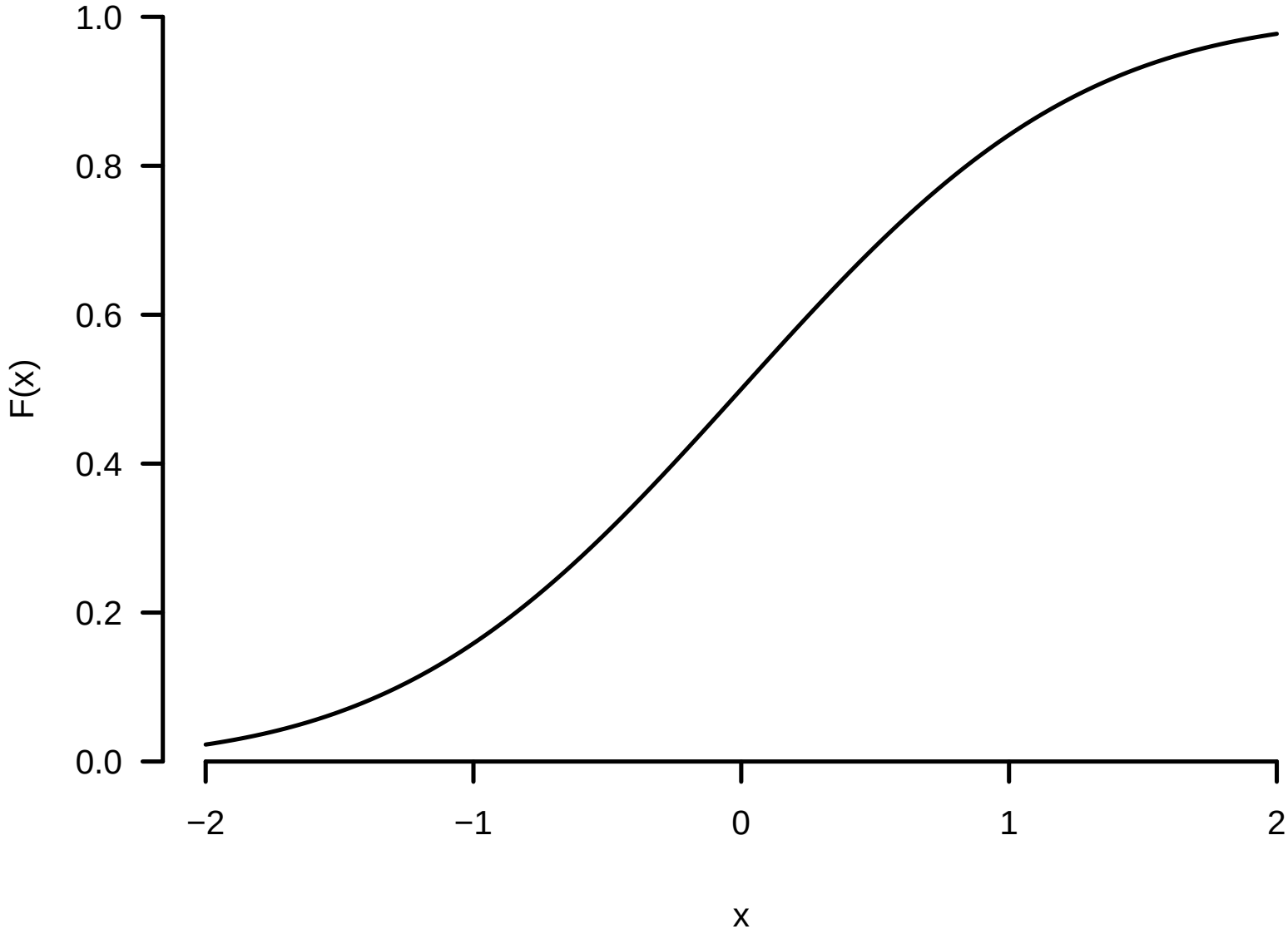
Quantiles divide a probability distribution into sections having equal probabilities.

For example,

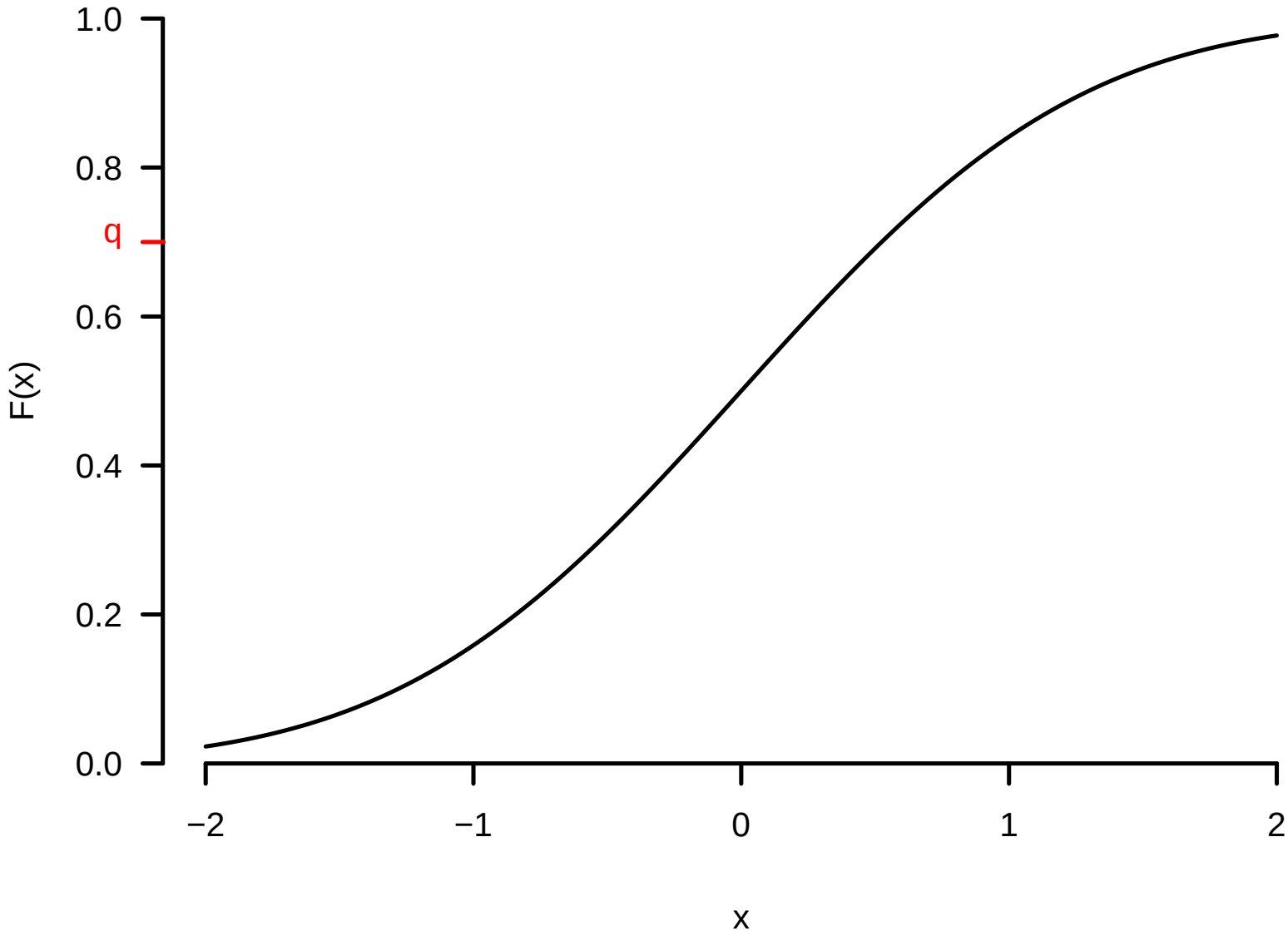
- the median divides the distribution in two
- quartiles divide the distribution in four
- deciles divide the distribution in 10
- percentiles divide the distribution in 100

The generic name for all these quantities is quantile.

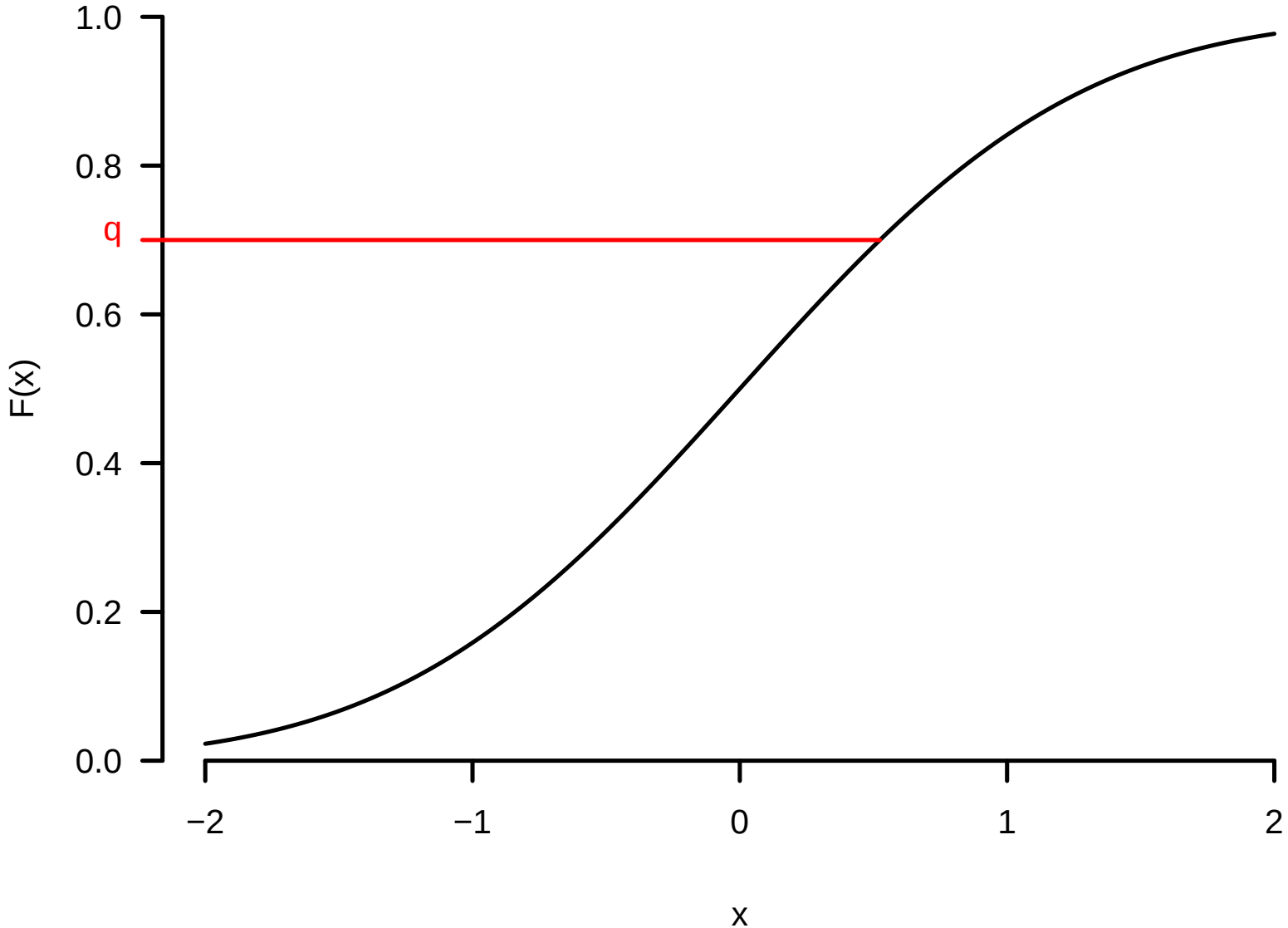
# Quantiles



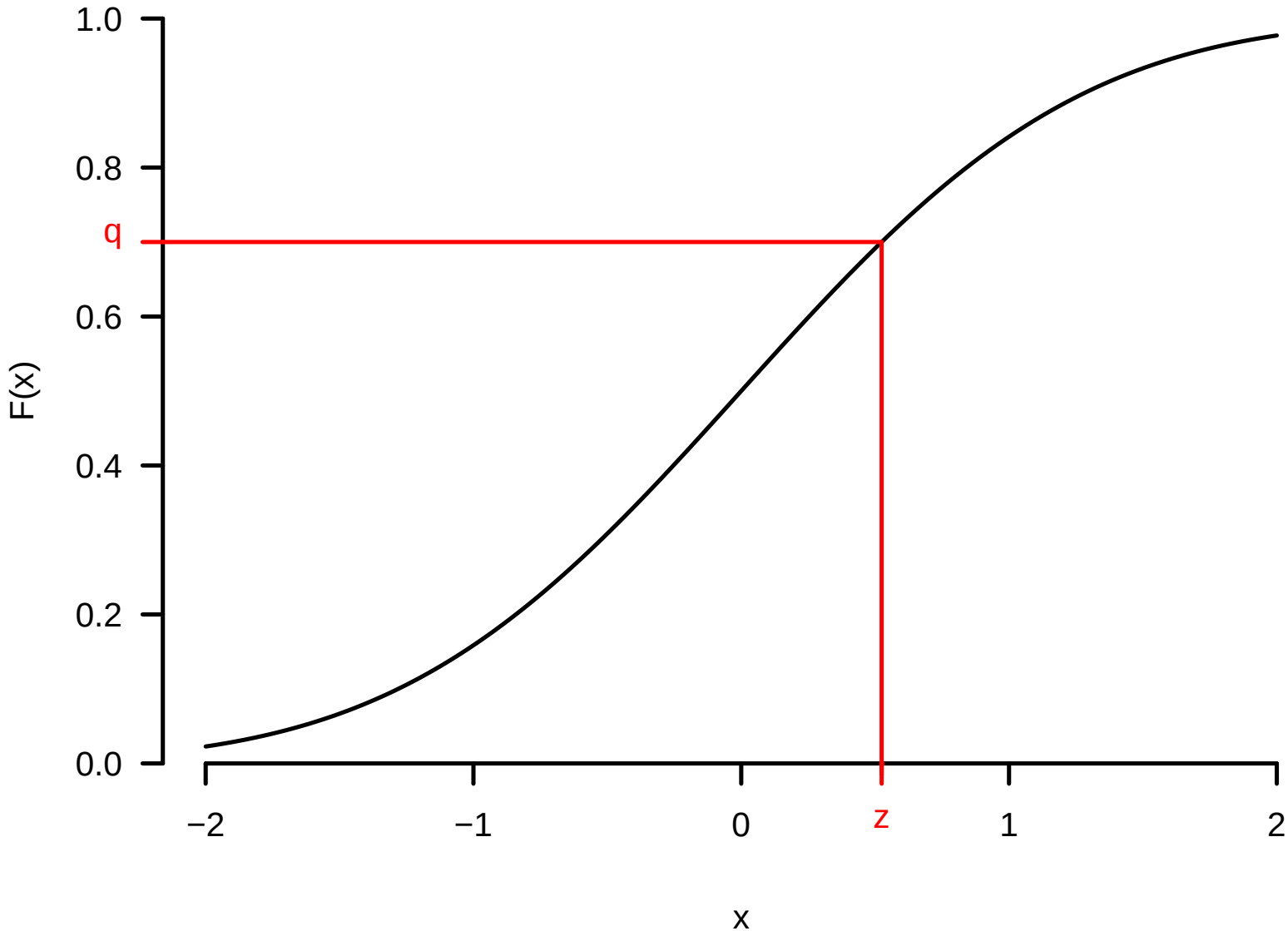
# Quantiles



# Quantiles



# Quantiles

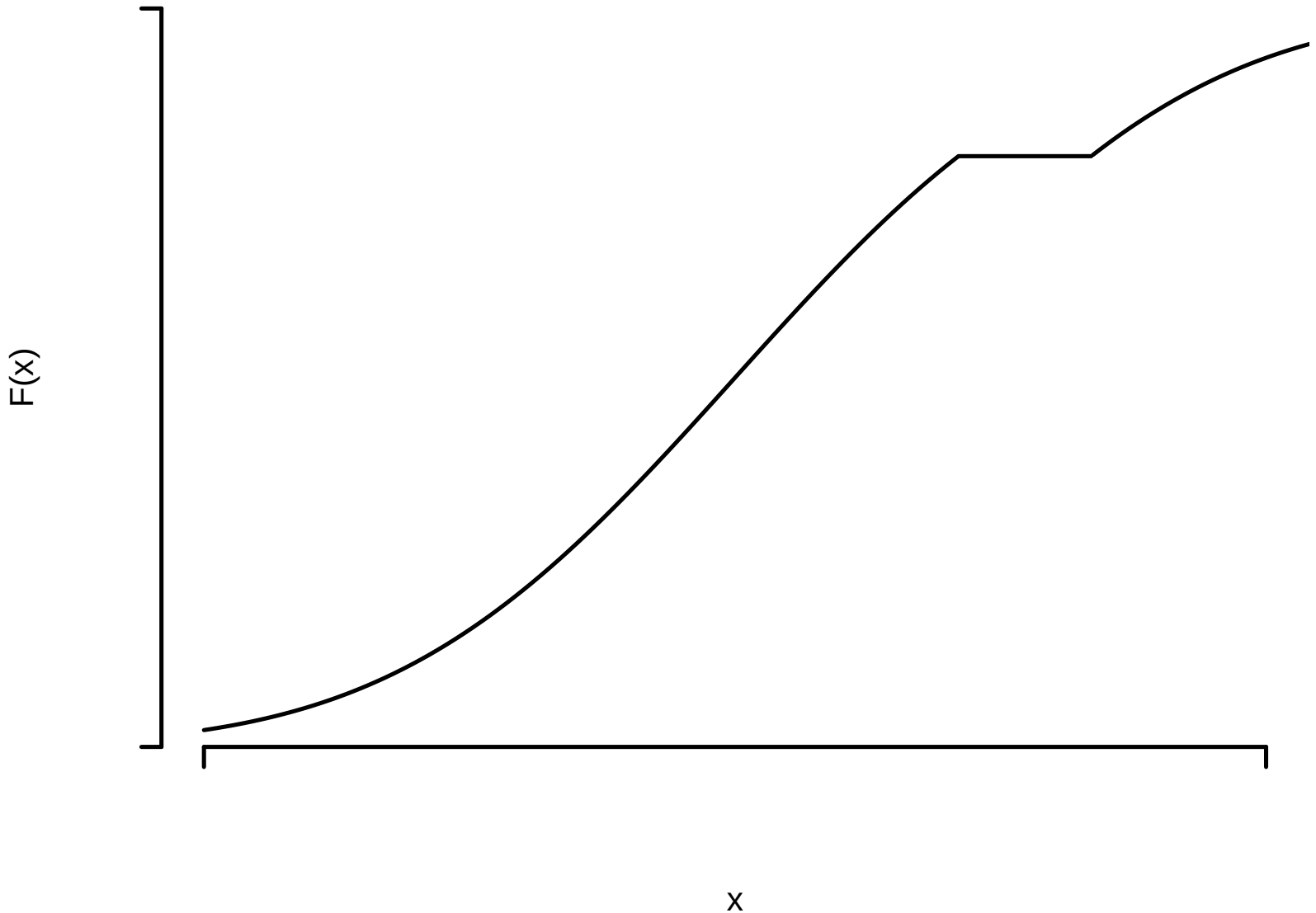


**Definition**

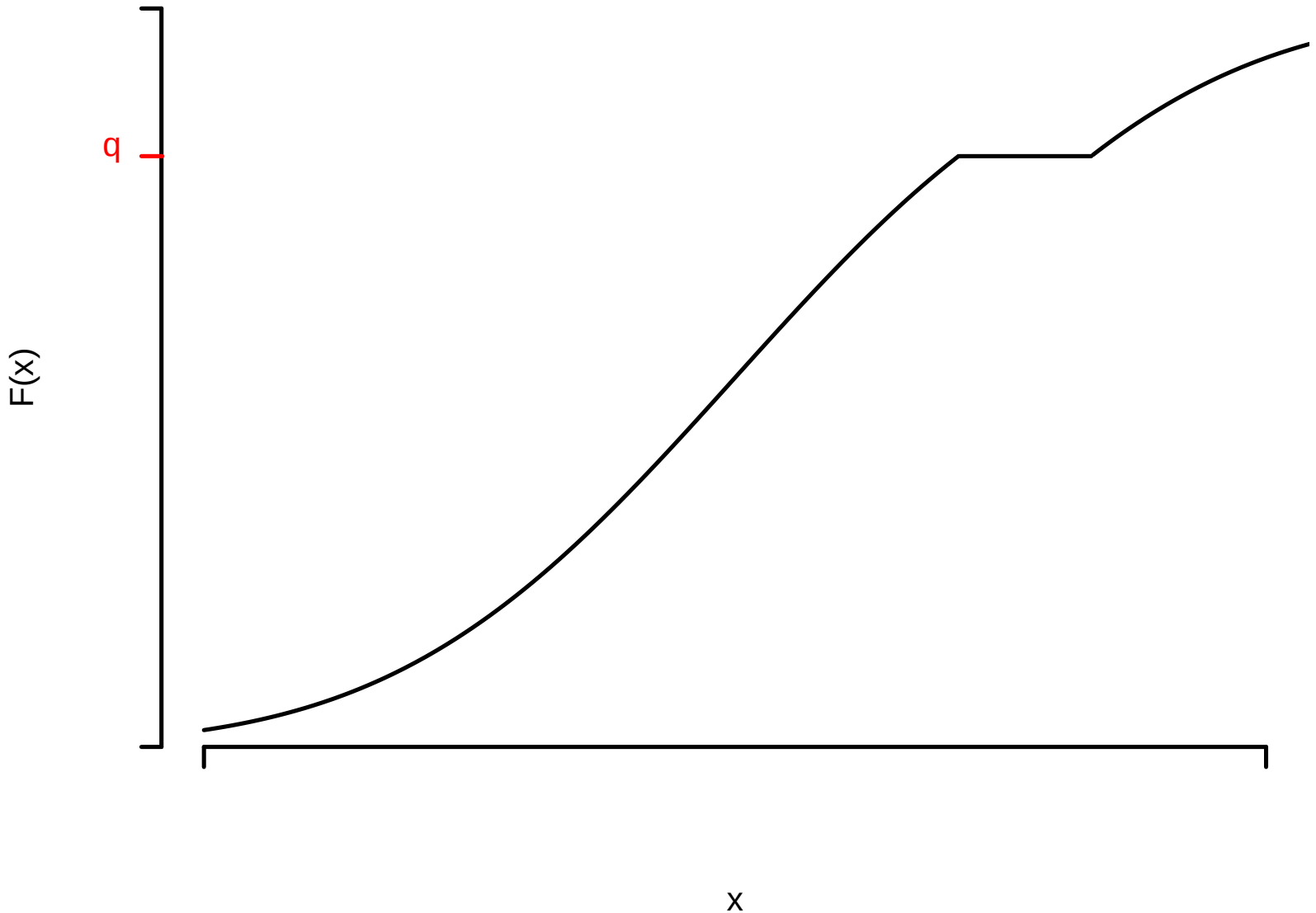
Given a distribution function $F(x)$ that is continuous and strictly increasing, for $0 < q < 1$, the $q$ quantile is the value $z$ such that a fraction $q$ of the distribution is to the left of $z$:

$$P(X \leq z) = q$$

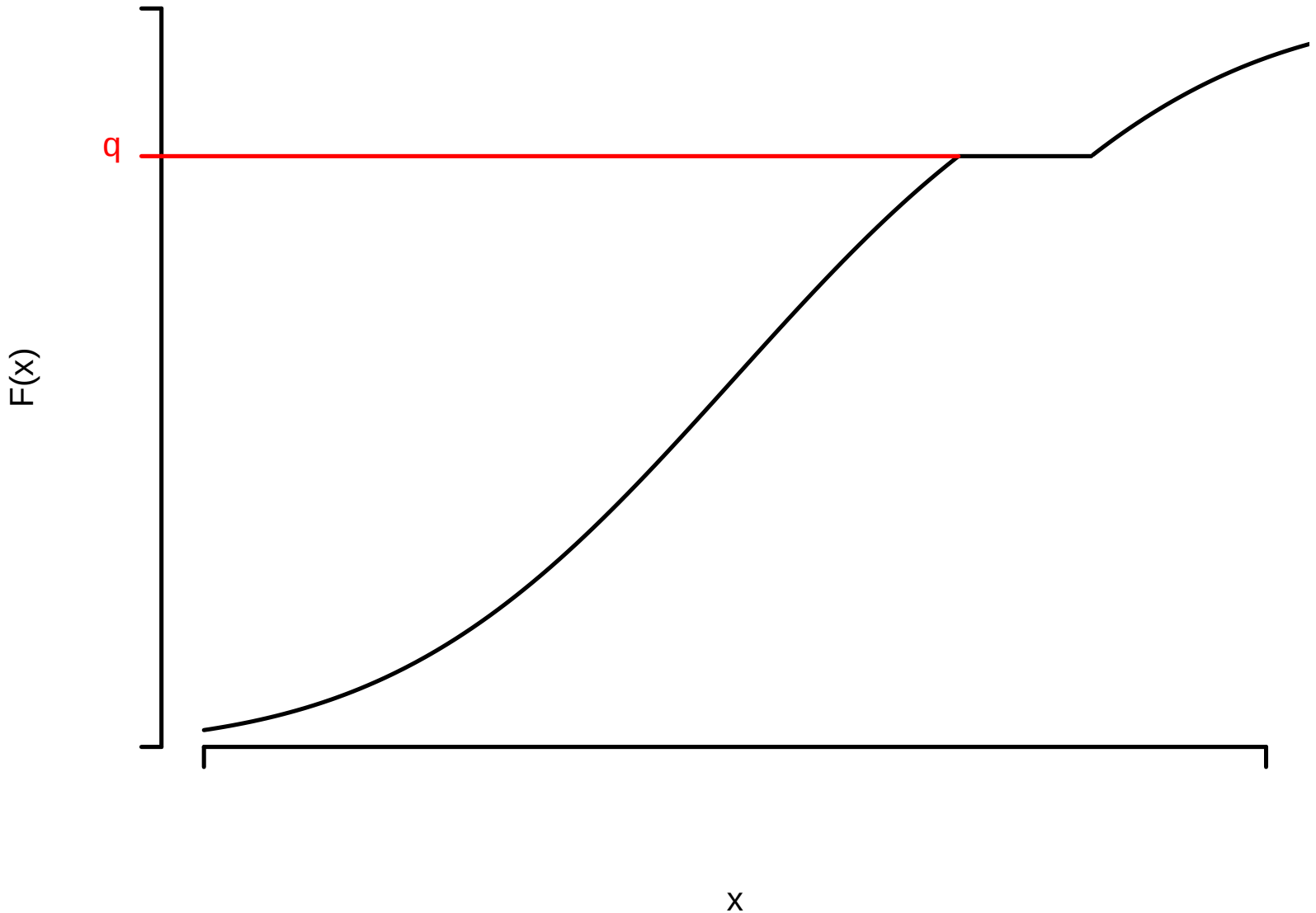
# Quantiles



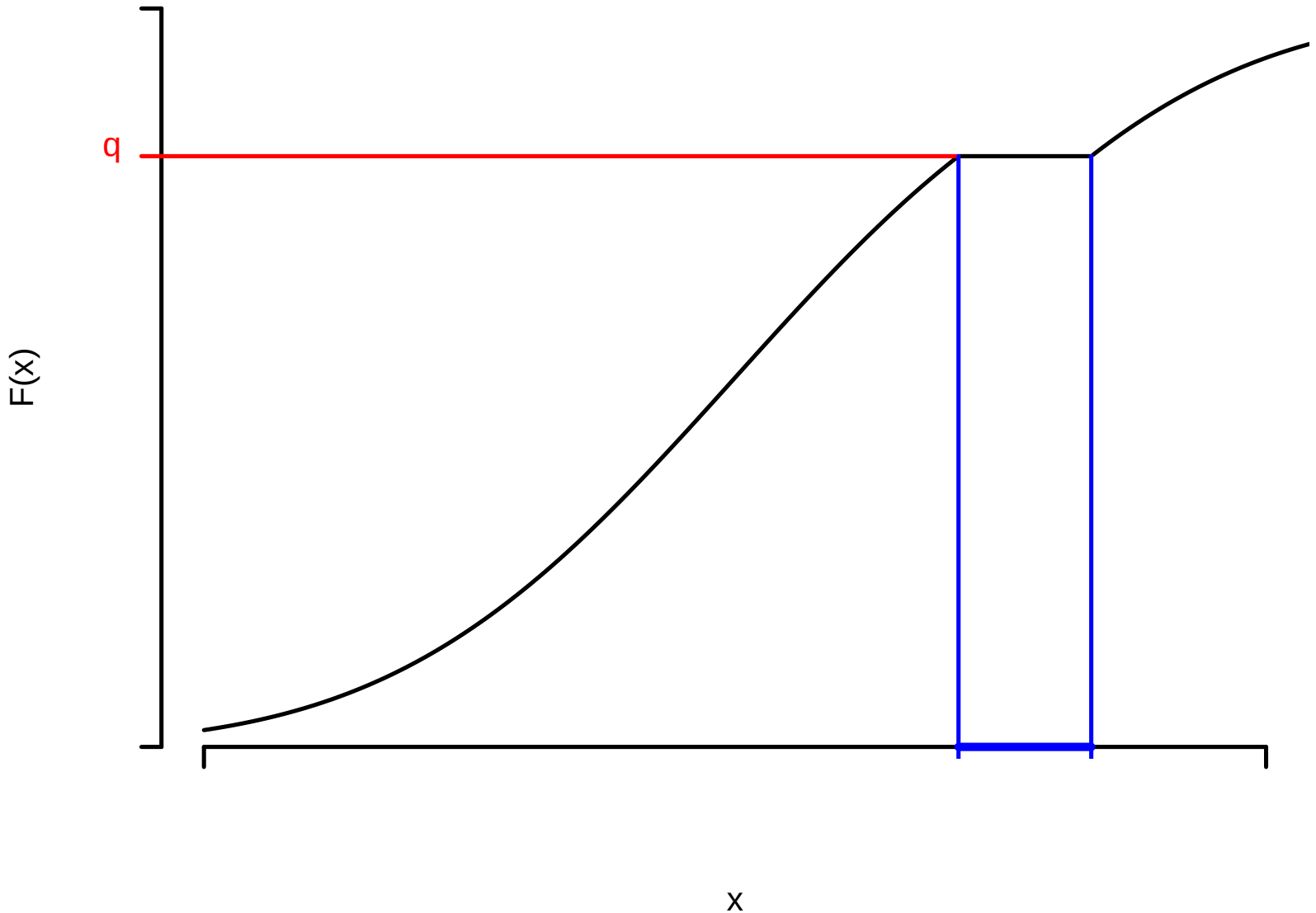
# Quantiles



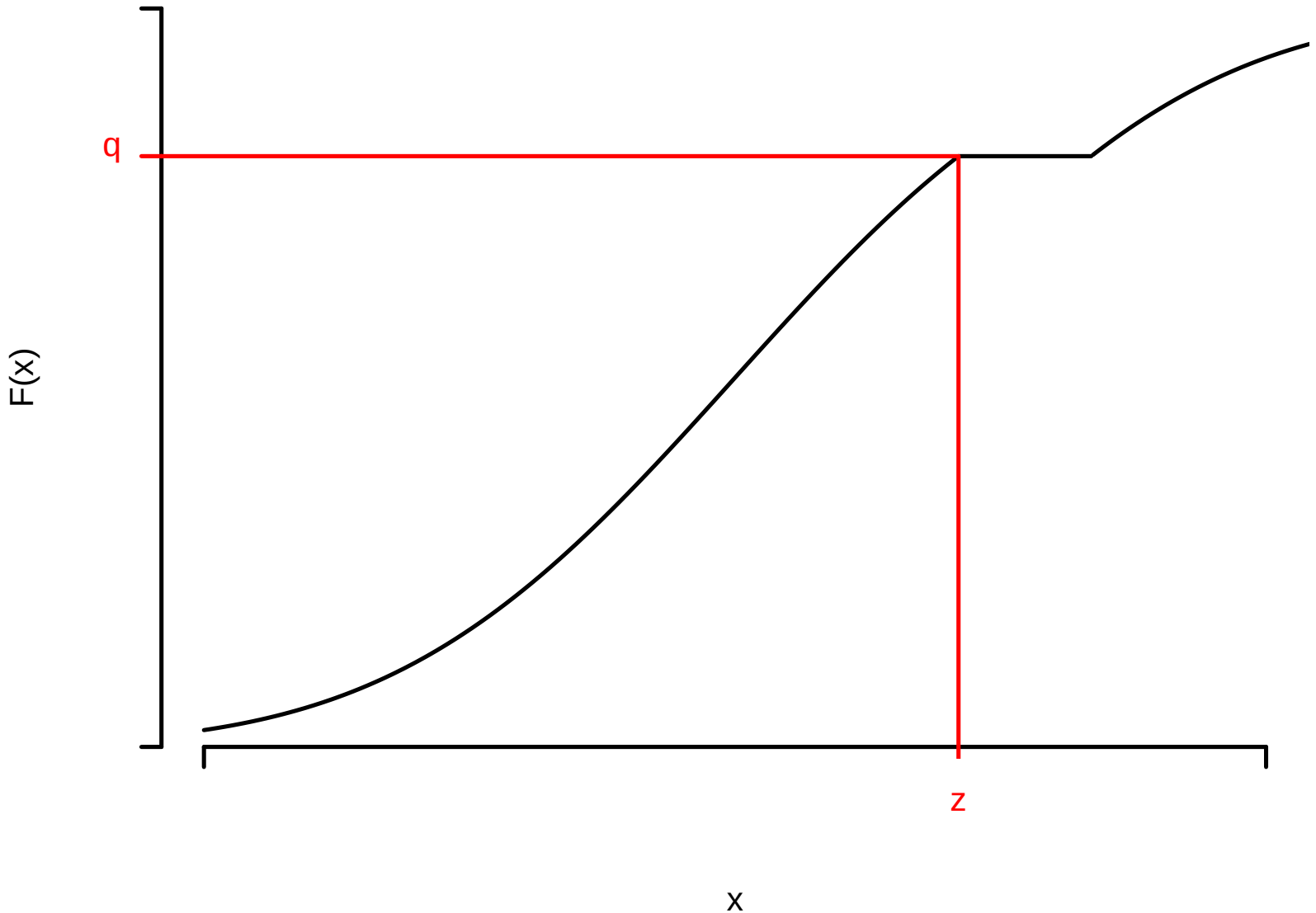
# Quantiles



# Quantiles



# Quantiles

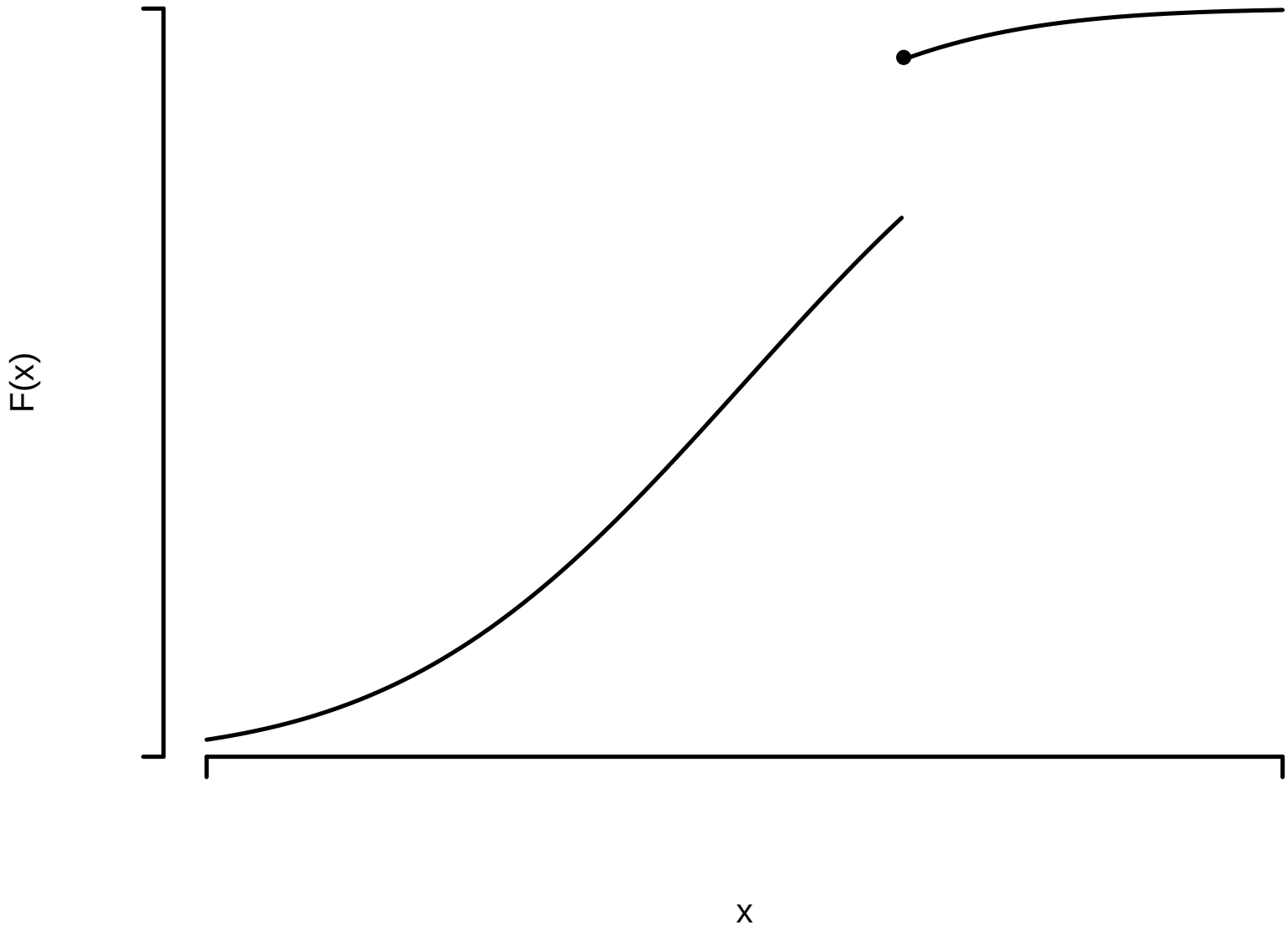


If the quantile is not unique, we take the smallest value for which $F(z) = q$:

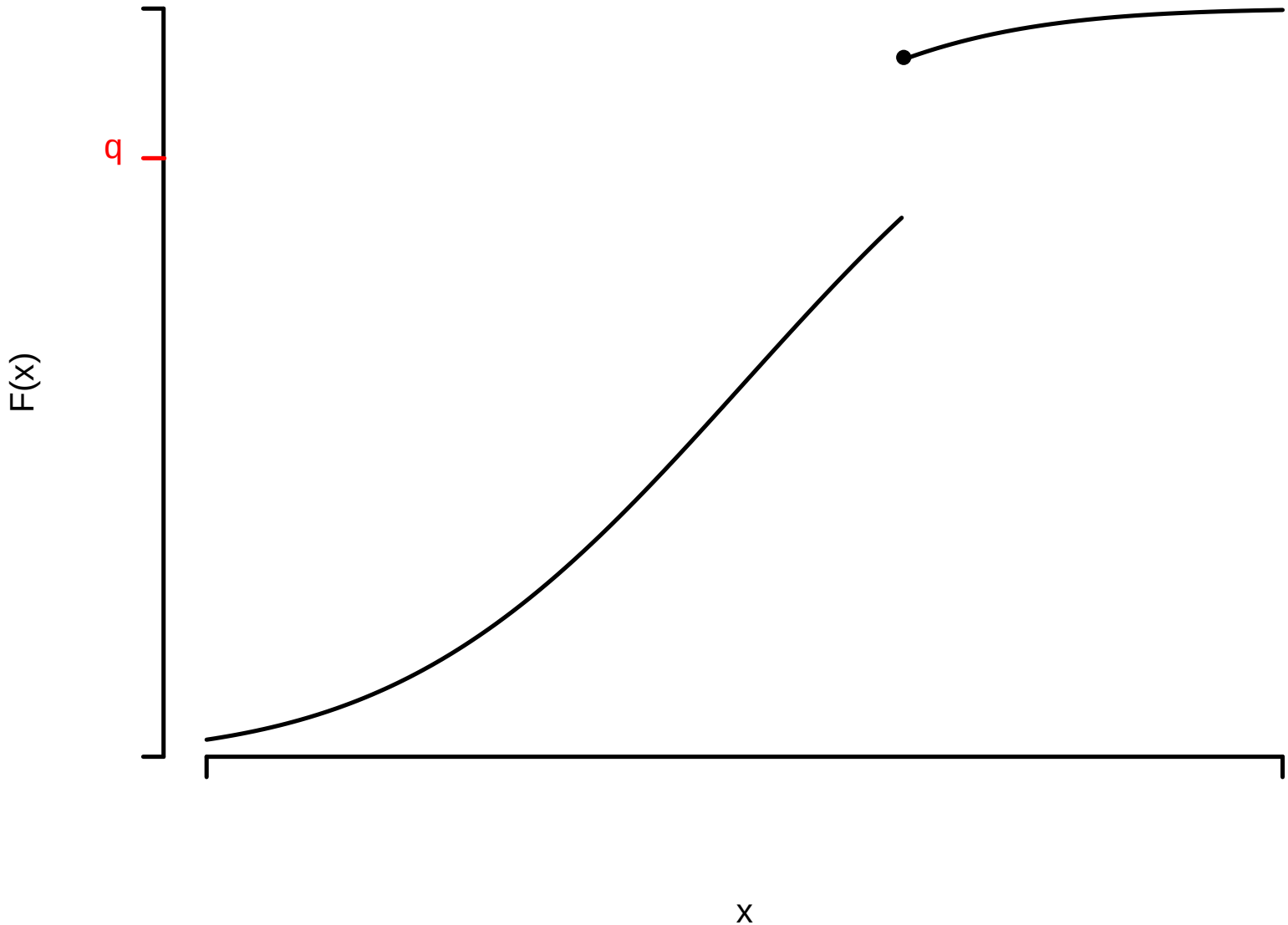
$$z = \inf\{x : F(x) \geq q\}$$

Note that if we are in the earlier case, i.e., the function is continuous and strictly increasing, this definition gives the same value for $z$ as the previous one.

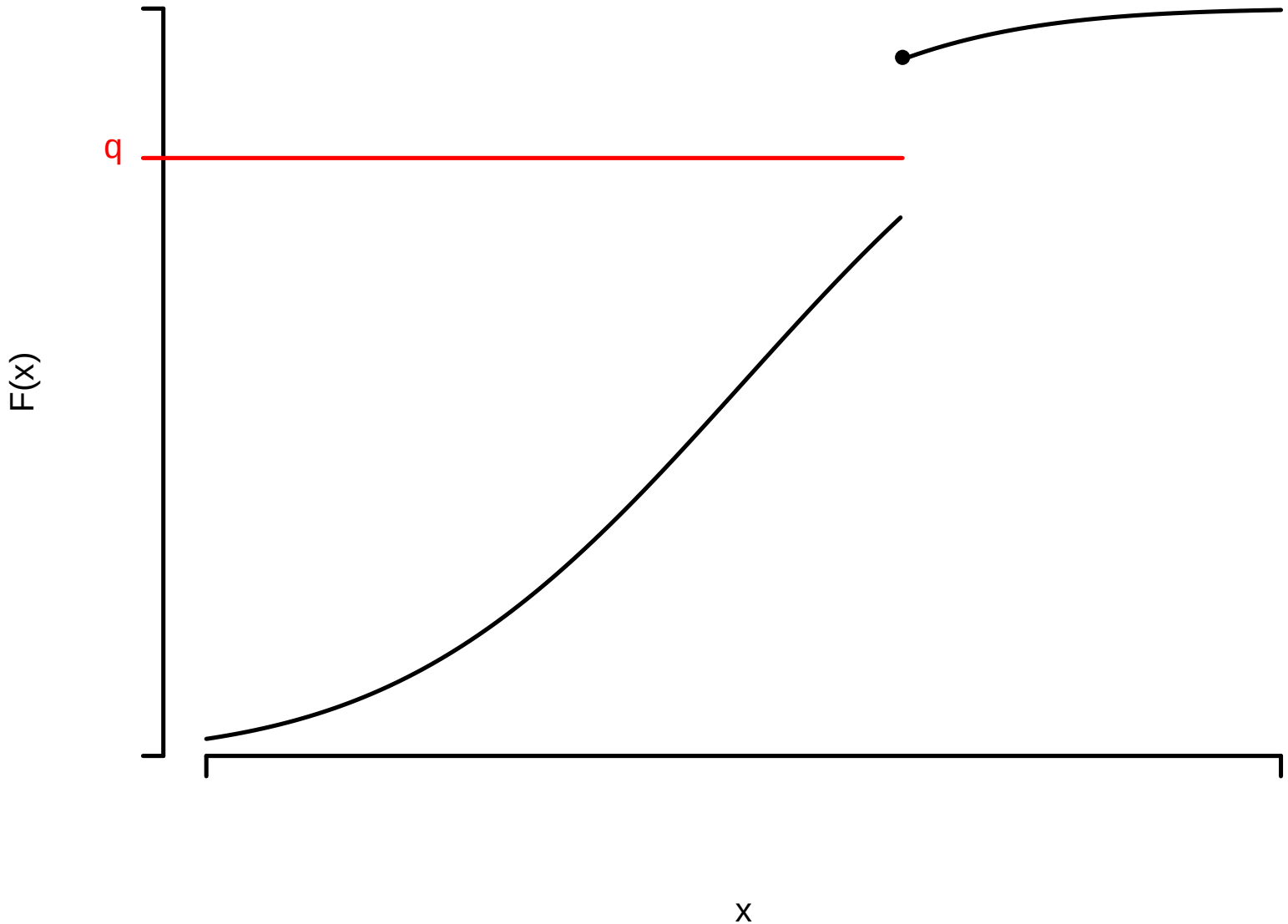
# Quantiles



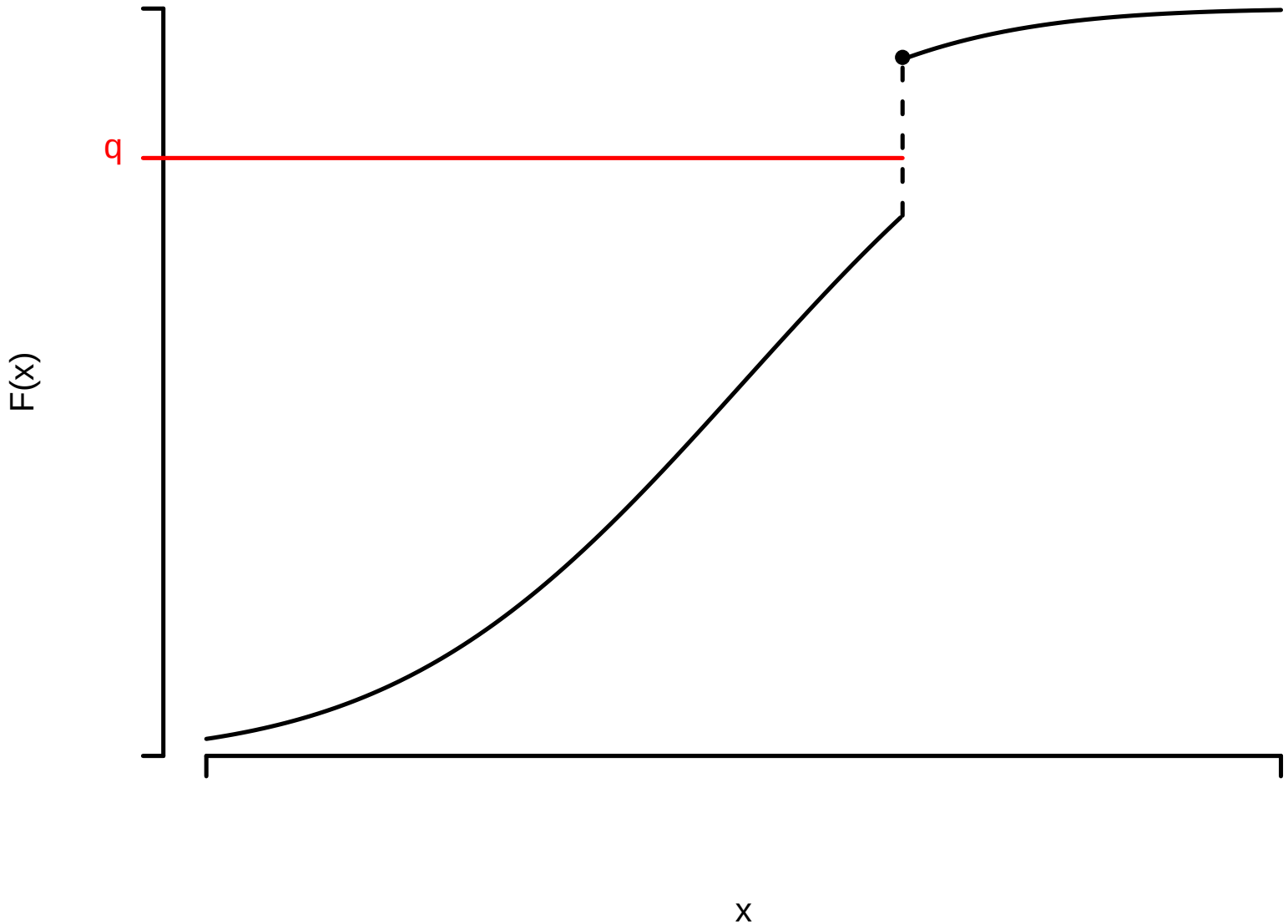
# Quantiles



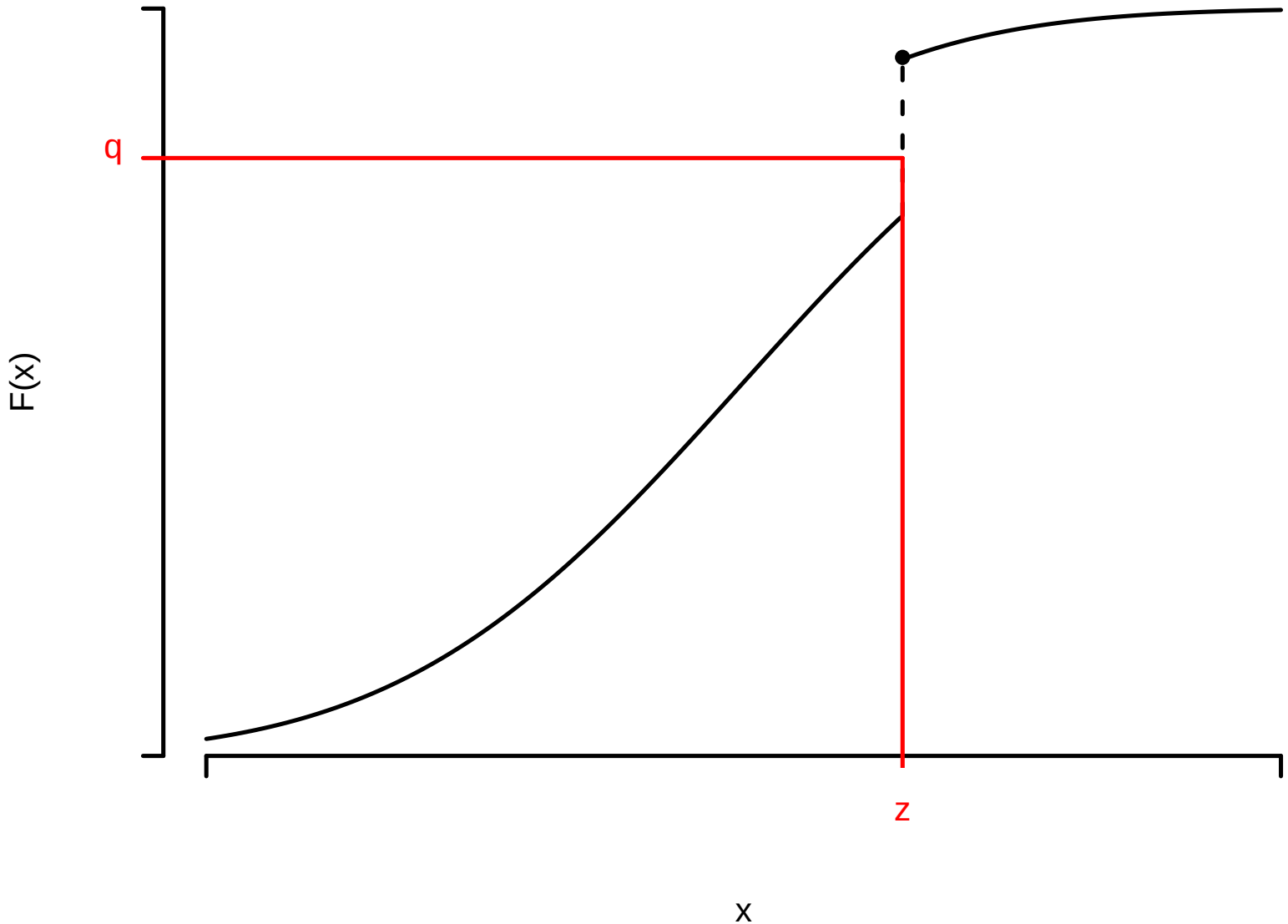
# Quantiles



# Quantiles



# Quantiles



If $F$ is discontinuous then it may happen that there is no value of $z$ that satisfies $F(z) = q$. In this case

$$z = \inf\{x : F(x) \geq q\}$$

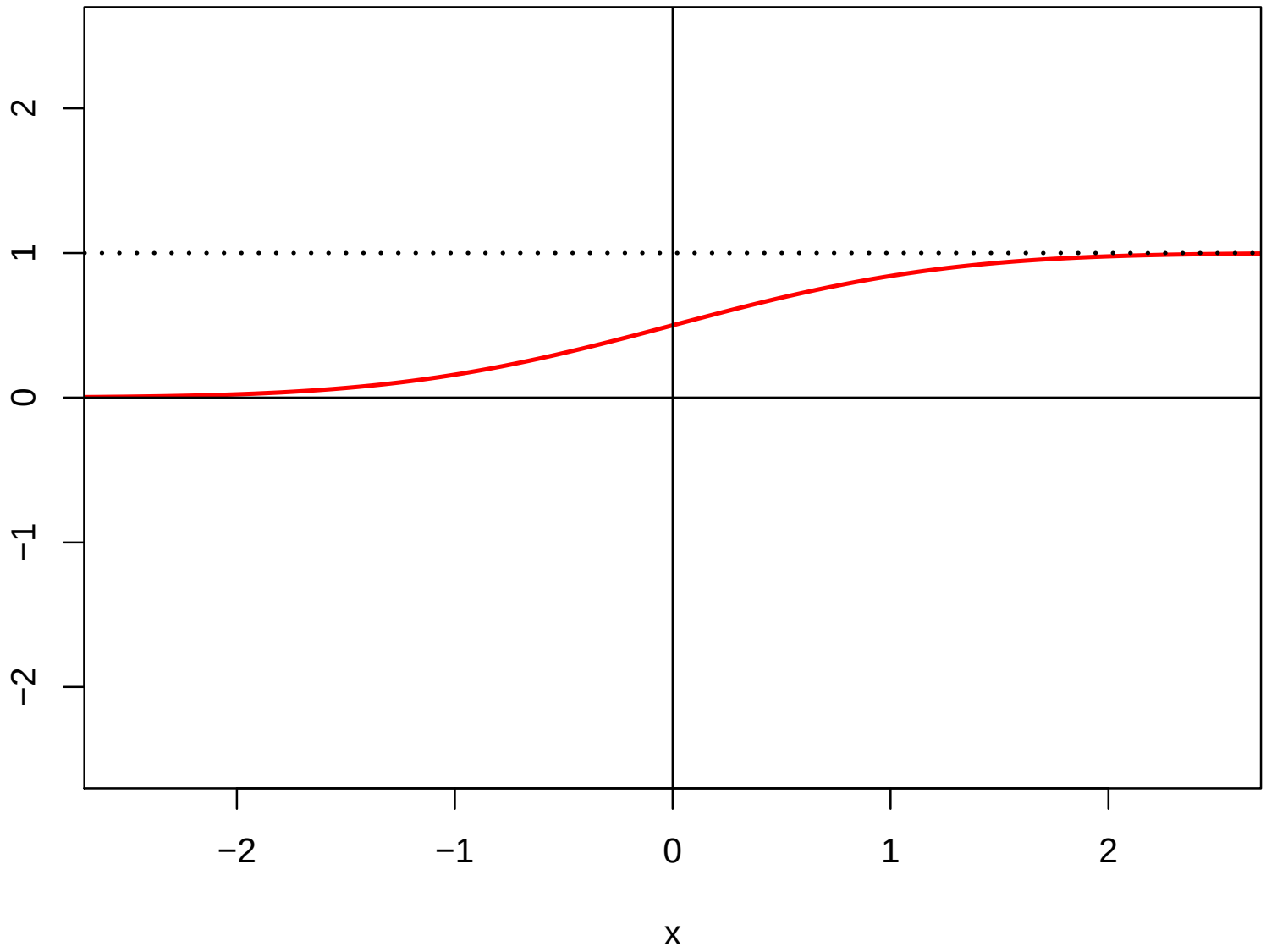
**Definition**

The **quantile function** $Q$ is the function that, given $q$, $0 < q < 1$, produces the value $z = \inf\{x : F(x) \geq q\}$.

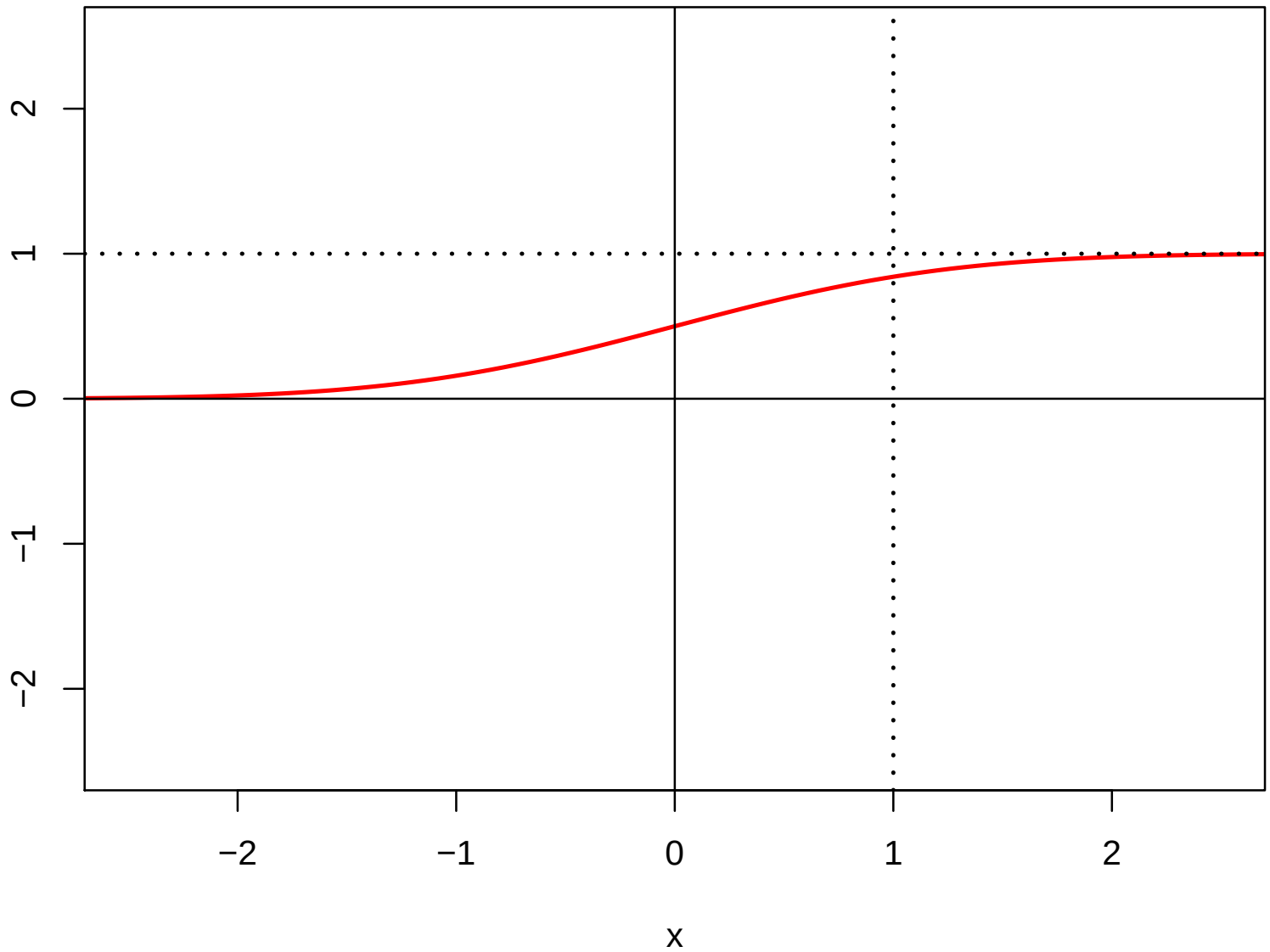
If $F$ is continuous and strictly increasing, then $Q$ is the inverse function of $F$.

The empirical quantiles are the quantiles of the empirical distribution.

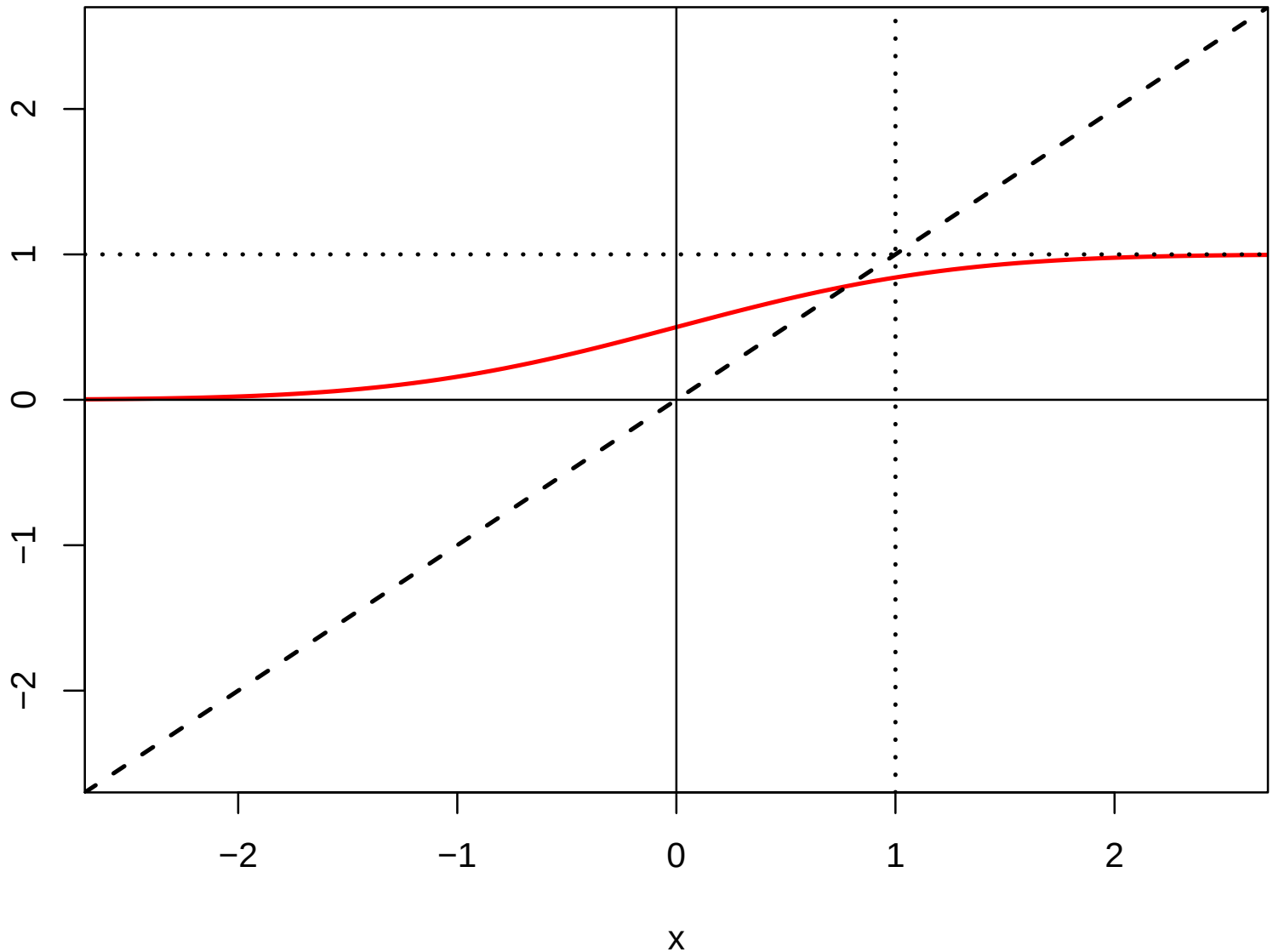
# Quantile function



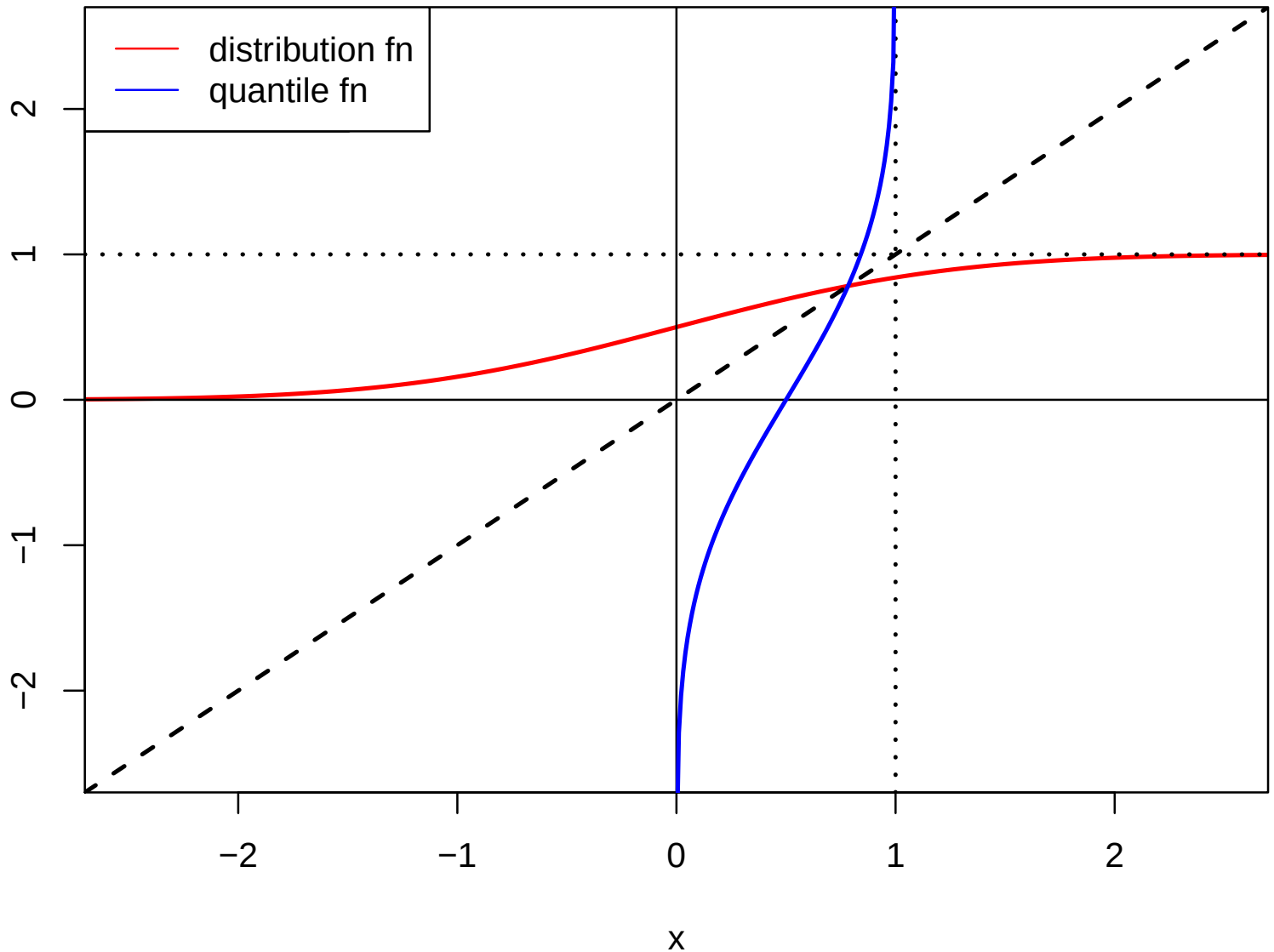
# Quantile function



# Quantile function



# Quantile function



# Quantile plots

The quantile plots, proposed by Wilk and Gnanadesikan in 1968, are a visual tool to compare the distribution of two sets of data or to compare a set of data with a reference distribution.

If the two distributions belong to the same location and scale family, the graph will be approximately a straight line.

# Quantile plots

Suppose we have two samples of the same size, $x_i, y_i, 1 \leq i \leq n$. The order statistics of the samples are the ordered values: For the $x$ sample, assuming there are no ties, this would be

$$x_{(1)} < x_{(2)} < \cdots < x_{(n-1)} < x_{(n)}$$

The quantile plot for the two samples is the plot of ordered values of $x$ versus the ordered values of $y$, if both samples have the same size. If the two samples are not the same size, linear interpolation is used.

In `R` the function for making quantile plots to compare two samples is `qqplot`

# Quantile plots

When we want to compare with a reference distribution, the empirical quantiles are plotted against the quantiles calculated from the reference distribution.

In particular, the function `qqnorm` in `R` draws a quantile plot to compare a given data set with the normal distribution.

If the fit is good, the points should appear to be on a straight line.

# Quantile plots