10. Probability distributions

These notes consider the Chapter 10 of the handbook on various probability distributions.

Discrete distributions

For random variables with countable number of possible values.

Binomial distribution

See page 200 for proper definition.

Useful when a random variable X has exactly two exclusive possible outcomes (e.g. success/fail) with known probabilities $p \in [0, 1]$ (success) and $q = 1 - p$ (fail). For $n \in \mathbb{N}$ trials with $k = 0, 1, 2, \dots, n$ successes

$$X \sim Bin(n, p)$$

Probability Mass Function (PMF)

$$\mathrm{P}(X = k) = \binom{n}{k} p^k q^{n-k}$$

Cumulative Distribution Function (CDF)

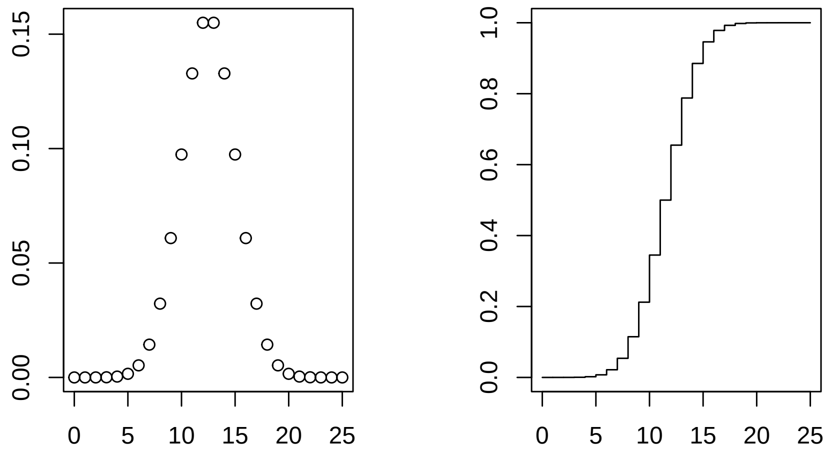
$$\mathrm{F}(k) = \mathrm{P}(X \leqslant k) = \sum_{i=0}^{k} \binom{n}{i} p^i q^{n-i}$$

In the following code examples, equal probabilities are assumed, $n =$ **size**, $p =$ **prob** $= 0.5$.

```r
# Sequence for visualization
binomial_seq <- seq(0, 25, by = 1)

# Functions
binomial_pmf <- dbinom(x = binomial_seq, size = 25, prob = 0.5)
binomial_cdf <- pbinom(q = binomial_seq, size = 25, prob = 0.5)

# Plot
par(mfrow = c(1, 2))
plot(binomial_seq, binomial_pmf, ann = FALSE)
plot(binomial_seq, binomial_cdf, type = "S", ann = FALSE)
```




```r
# Probability for exactly 3 successes out of 10 trials
dbinom(x = 3, size = 10, prob = 0.5)
```

```
## [1] 0.1171875
```

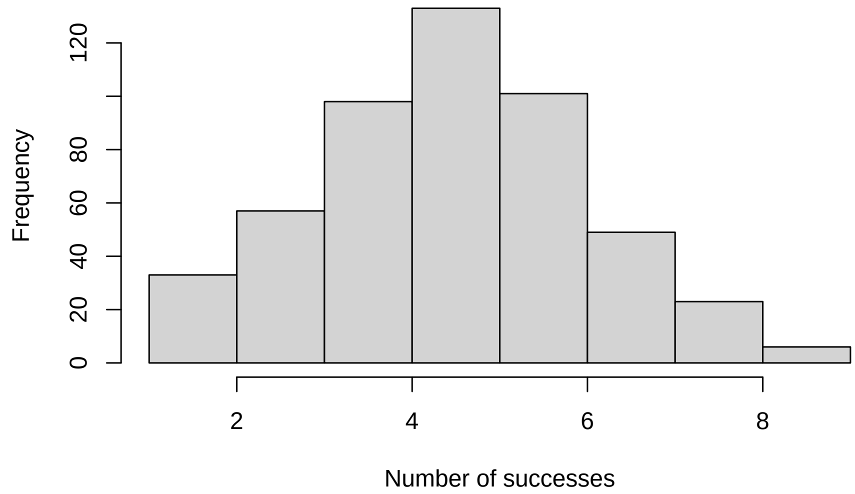
```r
# Probability for up to 3 successes out of 10 trials
pbinom(q = 3, size = 10, prob = 0.5)
```

```
## [1] 0.171875
```

```r
# Simulate 10 times how many successes there is using random numbers
rbinom(n = 10, size = 10, prob = 0.5)
```

```
##  [1] 2 6 6 6 6 6 1 4 2 7
```

```r
# With large enough n, expected value (np = 5) should become visible
rbinom500 <- rbinom(n = 500, size = 10, prob = 0.5)
hist(rbinom500, xlab = "Number of successes", ylab = "Frequency", main = NULL)
```




```r
summary(rbinom500)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   4.000   5.000   4.954   6.000   9.000
```

Poisson distribution

See page 204.

Useful when estimating amounts in random processes where the expected value (λ) is known

$$X \sim Poi(\lambda)$$

PMF

$$\mathrm{P}(X=k) = \frac{\lambda^k e^{-\lambda}}{k!}\ ,\ k = 0, 1, 2, \ldots, n$$

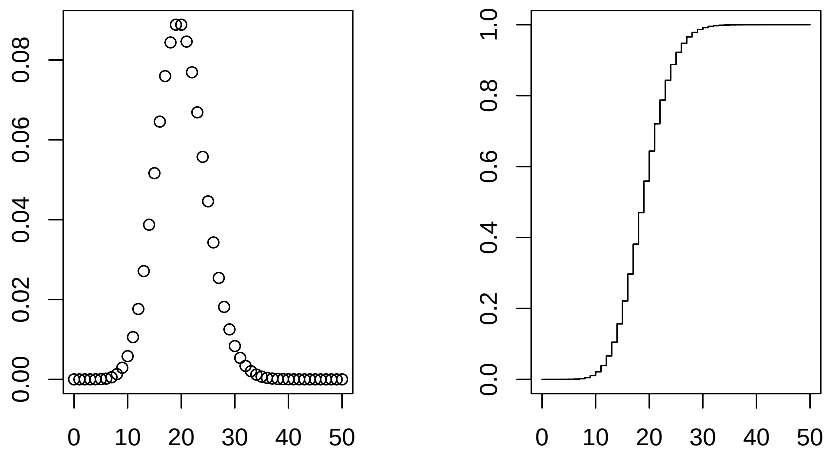
Note that the Poisson distribution can be used to approximate the binomial distribution when n is large and p is small. In this case, $\lambda = np$ and

$$\mathrm{P}(X=k) = \frac{(np)^k e^{-np}}{k!}$$

```r
# Sequence for visualization
poisson_seq <- seq(0, 50, by = 1)

# Functions
poisson_pmf <- dpois(x = poisson_seq, lambda = 20)
poisson_cdf <- ppois(q = poisson_seq, lambda = 20)
```

```r
# Plot
par(mfrow = c(1, 2))
plot(poisson_seq, poisson_pmf, ann = FALSE)
plot(poisson_seq, poisson_cdf, type = "S", ann = FALSE)
```




```r
# A bridge is crossed by 12 people per minute, on average.
# What is the probability that exactly 16 people crosses it in a minute?
dpois(x = 16, lambda = 12)
```

```
## [1] 0.05429334
```

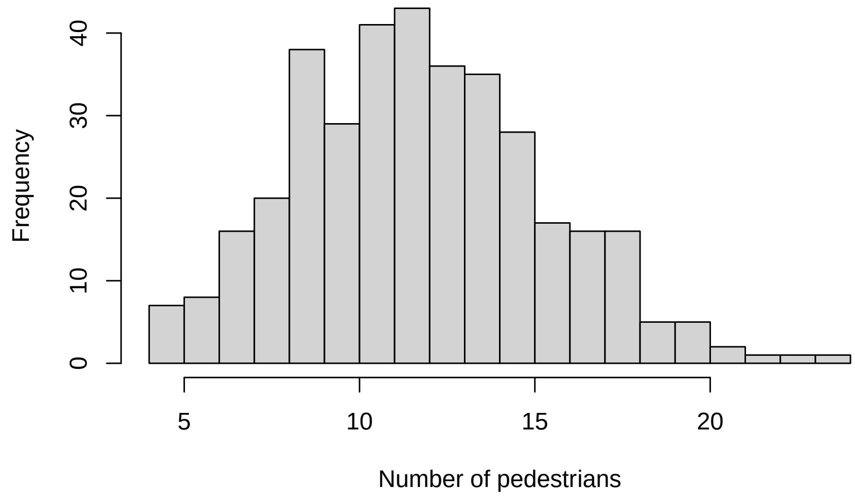
```r
# A bridge is crossed by 12 people per minute, on average.
# What is the probability that up to 16 people crosses it in a minute?
ppois(q = 16, lambda = 12)
```

```
## [1] 0.898709
```

```r
# Simulate amount of people per minute 10 times
rpois(n = 10, lambda = 12)
```

```
##  [1] 16 14  6 12 13 15 11 15 15 15
```

```r
# By repeating the observation once every day for a year,
# the expected value (lambda = 12) should become visible
poisson_pedestrians <- rpois(n = 365, lambda = 12)
hist(poisson_pedestrians, breaks = 20,
     xlab = "Number of pedestrians", ylab = "Frequency", main = NULL)
```




```r
summary(poisson_pedestrians)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    4.00   10.00   12.00   12.24   15.00   24.00
```

Continuous distributions

For random variables with infinite number of possible values.

Generic definitions

See page 208.

The probability that an event happens between an interval $[a, b]$ can be calculated from the Probability Density Function (PDF) $f(x)$ as an integral

$$\mathrm{P}(a \leqslant X \leqslant b) = \int_a^b f(x) \, \mathrm{d}x$$

Note that the point probabilities for continuous random variables are intrinsically zero

$$\int_a^a f(x) \, \mathrm{d}x = 0$$

CDF

$$\mathrm{P}(X \leqslant t) = \int_{-\infty}^t f(x) \, \mathrm{d}x$$

Normal distribution

See page 210.

Many phenomena in nature follow the normal distribution. Additionally, so called central limit theorem states that the averages of idenpendet random samples drawn from a population approximately follow the normal distribution - even if the population is better desrcibed by some other distribution!

Normal distribution is defined by expected value (and median and mode due to symmetry) μ and (standard) deviation σ.

$$X \sim N(\mu, \sigma^2)$$

PDF

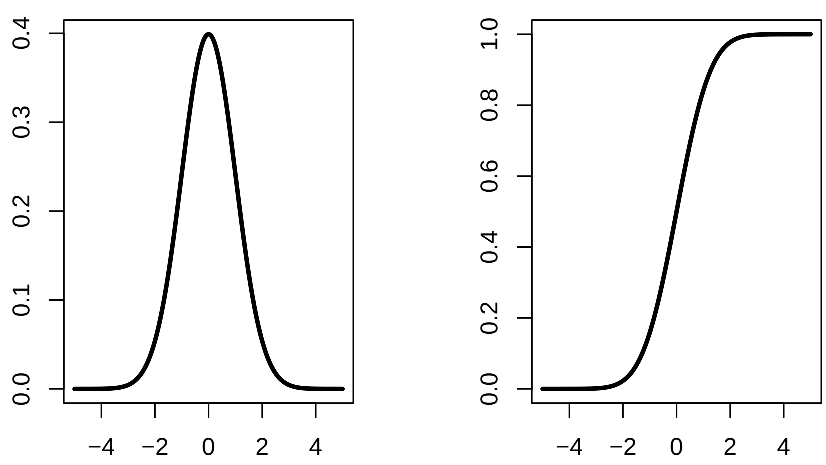
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

In the following code examples, $\mu =$ **mean**, $\sigma =$ **sd**.

```r
# Sequence for visualization
normal_seq <- seq(-5, 5, by = 0.01)

# Functions
normal_pmf <- dnorm(x = normal_seq, mean = 0, sd = 1)
normal_cdf <- pnorm(q = normal_seq, mean = 0, sd = 1)

# Plot
par(mfrow = c(1, 2))
plot(normal_seq, normal_pmf, type = "l", lwd = 3, ann = FALSE)
plot(normal_seq, normal_cdf, type = "l", lwd = 3, ann = FALSE)
```



```r
# The average length of a zebrafish is 22 mm with a standard diviation of 1 mm.
# How likely is a fish 23 mm long?
dnorm(x = 23, mean = 22, sd = 1)
```

```
## [1] 0.2419707
```

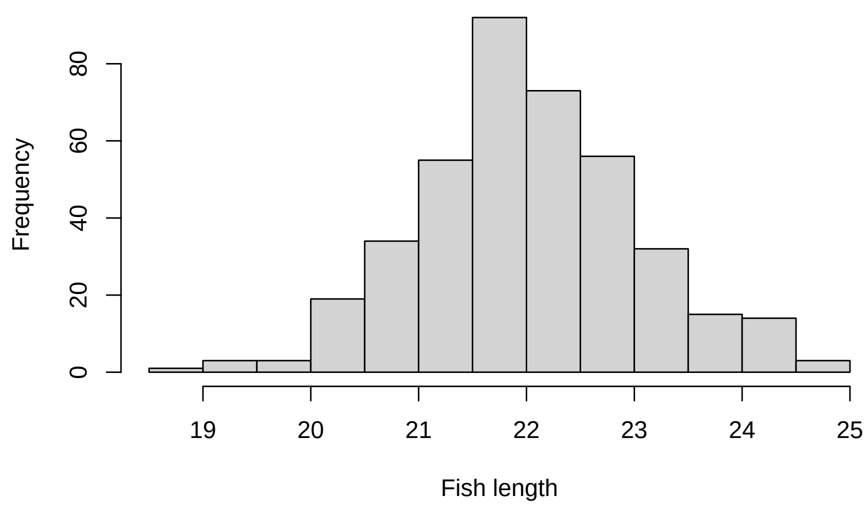
```r
# What fraction of the fish are up to 23 mm long?
pnorm(q = 23, mean = 22, sd = 1)
```

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

```r
# Simulate length of five random fish
rnorm(n = 5, mean = 22, sd = 1)
```

```
## [1] 23.35507 21.42742 23.97557 21.72851 23.58276
```

```r
# With large enough n, expected value should become visible
normal_zebrafish <- rnorm(n = 400, mean = 22, sd = 1)
hist(normal_zebrafish, breaks = 20,
     xlab = "Fish length", ylab = "Frequency", main = NULL)
```




```r
summary(normal_zebrafish)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.78   21.35   21.96   22.01   22.64   24.93
```

Student’s t-distribution

See page 215.

Similar to the normal distribution, Student’s t distribution can be used to estimate the expected value of a population using large enough amout of sample averages.

The Student’s t distribution is defined by just one parameter, namely the degrees of freedom (DoF) ν. Compared to the normal distribution, the advantage here is that prior knowledge of parameters μ and σ is not needed - which oftentimes is the case. Additionally, with a large ν, the Student’s t distribution approaches the standard normal distribution N(0,1).

PDF is defined using the Gamma function $\Gamma(n) = (n-1)!$.

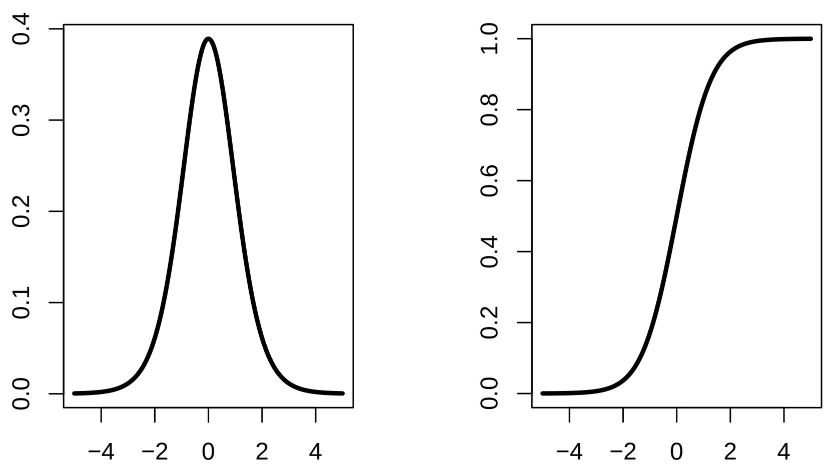
$$f(t) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi\nu}\Gamma(\frac{\nu}{2})}\left(1+\frac{t^2}{\nu}\right)^{-(\nu+1)/2}$$

In the following code examples, $\nu =$ **df**.

```
# Sequence for visualization
students_t_seq <- seq(-5, 5, by = 0.01)

# Functions
students_t_pmf <- dt(x = students_t_seq, df = 10)
students_t_cdf <- pt(q = students_t_seq, df = 10)

# Plot
par(mfrow = c(1, 2))
plot(normal_seq, students_t_pmf, type = "l", lwd = 3, ann = FALSE)
plot(normal_seq, students_t_cdf, type = "l", lwd = 3, ann = FALSE)
```



```r
# Same as in the earlier examples
dt(x = 2, df = 20)
```

```
## [1] 0.05808722
```

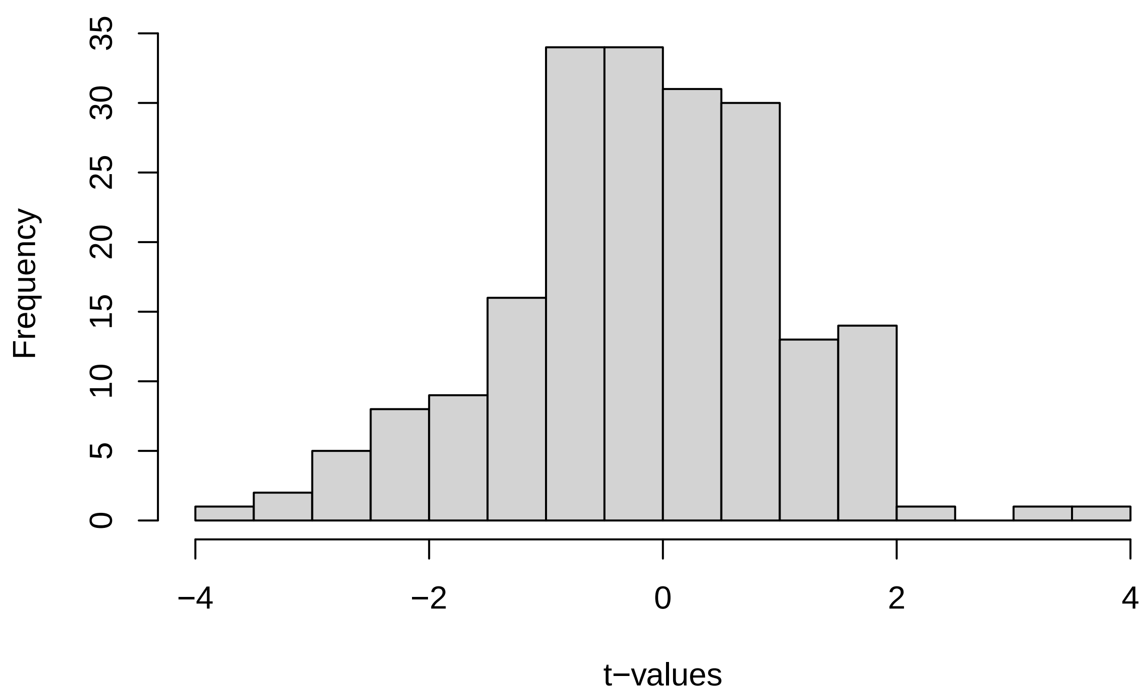
```r
pt(q = 2, df = 20)
```

```
## [1] 0.9703672
```

```r
rt(n = 5, df = 20)
```

```
## [1]  1.8902778 -0.9411542  0.2514571  0.2731861 -0.2830450
```

```r
# The mean should be around 0 regardless of DoF
student_values <- rt(n = 200, df = 5)
hist(student_values, breaks = 20,
     xlab = "t-values", ylab = "Frequency", main = NULL)
```




```r
summary(student_values)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.9990 -0.8367 -0.1295 -0.1720  0.6384  3.7014
```

Chi-squared distribution

See page 217.

The χ^2 distribution can be used to estimate the distribution of population variances. Suppose we have independent standard normal ($N(0,1)$) random variables $X_1, X_2, \ldots, X_k$, then the random variable

$$Q = \sum_{i=1}^{k} \chi_i^2$$

follows the χ^2 distribution ($Q \sim \chi^2(k)$) with $k-1$ DoF.

PDF

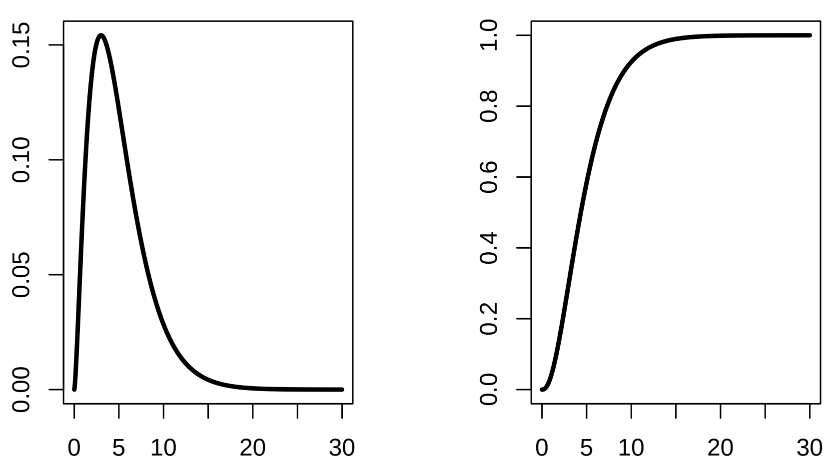
$$f(x;\, k) = \begin{cases} \dfrac{x^{k/2-1}e^{-x/2}}{2^{k/2}\Gamma\left(\frac{k}{2}\right)}, & x > 0; \\ 0, & \text{otherwise.} \end{cases}$$

In the following code examples, **df** is the DoF.

```r
# Sequence for visualization
chisq_seq <- seq(0, 30, by = 0.02)

# Functions
chisq_pmf <- dchisq(x = chisq_seq, df = 5)
chisq_cdf <- pchisq(q = chisq_seq, df = 5)

# Plot
par(mfrow = c(1, 2))
plot(chisq_seq, chisq_pmf, type = "l", lwd = 3, ann = FALSE)
plot(chisq_seq, chisq_cdf, type = "l", lwd = 3, ann = FALSE)
```



```r
# Same as in the earlier examples
dchisq(x = 2, df = 4)
```

```
## [1] 0.1839397
```

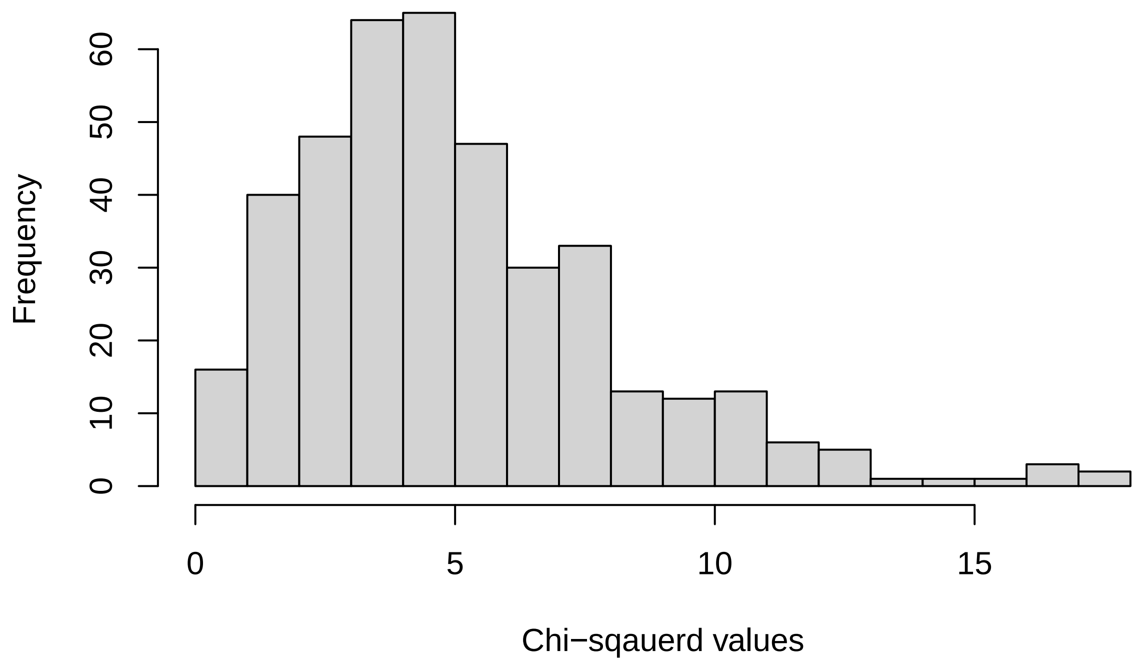
```r
pchisq(q = 2, df = 4)
```

```
## [1] 0.2642411
```

```r
rchisq(n = 5, df = 4)
```

```
## [1] 2.1620175 1.6149888 4.3746883 1.4707055 0.5914959
```

```r
# The mean should be around the DoF
chisq_values <- rchisq(n = 400, df = 5)
hist(chisq_values, breaks = 20,
     xlab = "Chi-sqauerd values", ylab = "Frequency", main = NULL)
```




```r
summary(chisq_values)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.05959  2.88340  4.49128  5.04503  6.56786 17.87608
```

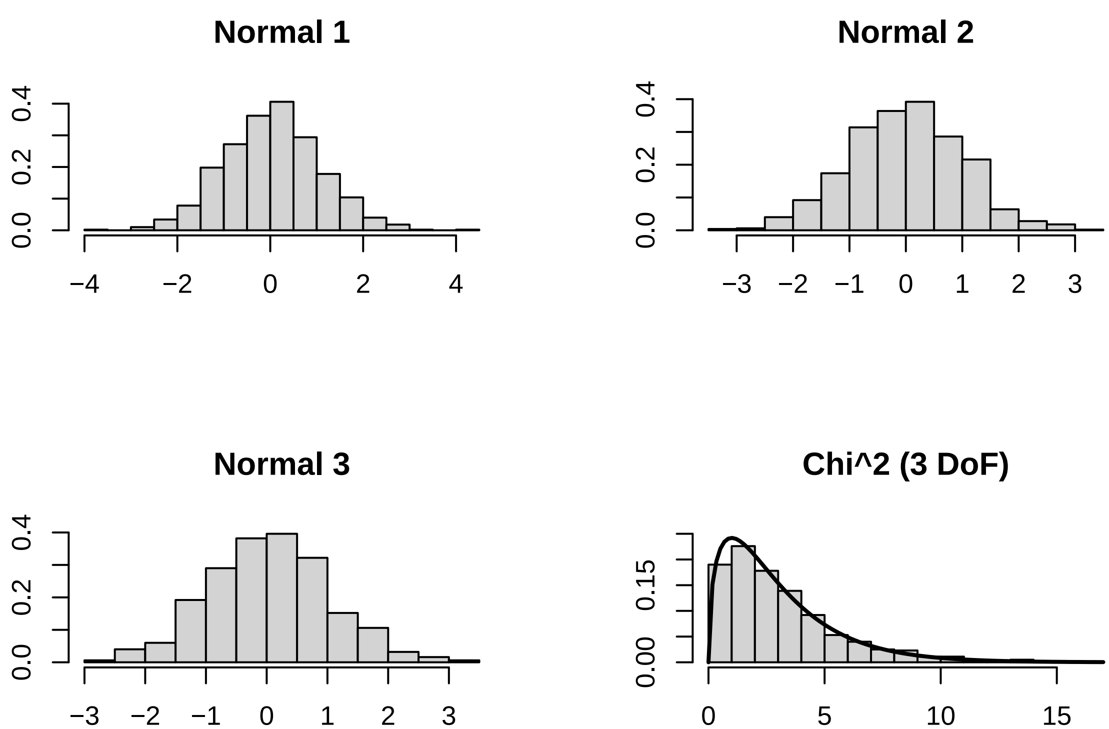
```r
# Let's also visualize the connection to the standard normal distribution
# First, generate some distributions
standard_normal_1 <- rnorm(n = 1000, mean = 0, sd = 1)
standard_normal_2 <- rnorm(n = 1000, mean = 0, sd = 1)
standard_normal_3 <- rnorm(n = 1000, mean = 0, sd = 1)

# Calculate the sum of squares (see random variable Q in the definition)
sum_of_squares = standard_normal_1^2 + standard_normal_2^2 + standard_normal_3^2

# Plots
par(mfrow = c(2, 2))
hist(standard_normal_1, breaks = 20,
     xlab = "", ylab = "", main = "Normal 1", prob = TRUE)
hist(standard_normal_2, breaks = 20,
     xlab = "", ylab = "", main = "Normal 2", prob = TRUE)
hist(standard_normal_3, breaks = 20,
     xlab = "", ylab = "", main = "Normal 3", prob = TRUE)

hist(sum_of_squares, breaks = 20, ylim = c(0, 0.25),
     xlab = "", ylab = "", main = "Chi^2 (3 DoF)", prob = TRUE)

# Overlap an ideal distribution with 3 DoF
curve(dchisq(x, df = 3), lwd = 2, add = TRUE)
```



Intuitively this should make sense: square of a standard normal distribution “flips” the negative values over the mean creating a skewed distribution and summing these enhances the effect.