Chapter 6
Distributions of One Variable

Distributions of one variable are called *univariate distributions*. They can be divided into *discrete distributions*, where the observations can only take on integer values (e.g., the number of children); and *continuous distributions*, where the observation variables are float values (e.g., the weight of a person).

The beginning of this chapter shows how to describe and work with statistical distributions. Then the most important discrete and continuous distributions are presented.

6.1 Characterizing a Distribution

6.1.1 Distribution Center

When we have a data sample from a distribution, we can characterize the center of the distribution with different parameters. Thereby the data can be evaluated in two ways:

1. By their value.
2. By their rank (i.e., their list-number when they are ordered according to magnitude).

a) Mean

By default, when we talk about the mean value we refer to the arithmetic mean $\bar{x}$:

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{6.1}$$

T. Haslwanter, *An Introduction to Statistics with Python*, Statistics and Computing,
DOI 10.1007/978-3-319-28316-6_6

Not surprisingly, the mean of an array x can be found with the command `np.mean`.

Real life data often include missing values, which are in many cases replaced by *nan*'s (*nan* stands for "Not-A-Number"). For statistics of arrays which include nan's, *numpy* has a number of functions starting with *nan*...

```
In [1]: import numpy as np

In [2]: x = np.arange(10)

In [3]: np.mean(x)
Out[3]: 4.5

In [4]: xWithNan = np.hstack( (x, np.nan) )    # append nan

In [5]: np.mean(xWithNan)
Out[5]: nan

In [6]: np.nanmean(xWithNaN)
Out[6]: 4.5
```

b) Median

The *median* is the value that comes half-way when the data are ranked in order. In contrast to the mean, the median is not affected by outlying data points. The median can be found with

```
In [7]: np.median(x)
Out[7]: 4.5
```

Note that when a distribution is symmetrical, as is the case here, the mean and the median value coincide.

c) Mode

The *mode* is the most frequently occurring value in a distribution.

The easiest way to find the mode value is the corresponding function in *scipy.stats*, which provides value and frequency of the mode value.

```
In [8]: from scipy import stats

In [9]: data = [1, 3, 4, 4, 7]

In [10]: stats.mode(data)
Out[10]: (array([4]), array([ 2.]))
```

d) Geometric Mean

In some situations the *geometric mean* can be useful to describe the location of a distribution. It can be calculated via the arithmetic mean of the log of the values.

$$mean_{geometric} = \left(\prod_{i=1}^{N} x_i\right)^{1/N} = exp\left(\frac{\sum_i ln(x_i)}{n}\right) \tag{6.2}$$

Again, the corresponding function is located in *scipy.stats*:

```
In [11]: x = np.arange(1,101)

In [12]: stats.gmean(x)
Out[12]: 37.992689344834304
```

Note that the input numbers for the geometric mean have to be positive.

6.1.2 Quantifying Variability

a) Range

The *range* is simply the difference between the highest and the lowest data value, and can be found with

```
range = np.ptp(x)
```

ptp stands for "peak-to-peak." The only thing that should be watched is outliers, i.e., data points with a value much higher or lower than the rest of the data. Often, such points are caused by errors in the selection of the sample or in the measurement procedure.

There are a number of tests to check for outliers. One of them is to check for data which lie more than 1.5*inter-quartile-range (IQR) above or below the first/third quartile ("quartiles" are defined in the next section).

b) Percentiles

The simplest way to understand *centiles*, also called *percentiles*, is to first define the *Cumulative Distribution Function* (*CDF*):

$$CDF(x) = \int_{-\infty}^{x} PDF(x)dx \tag{6.3}$$

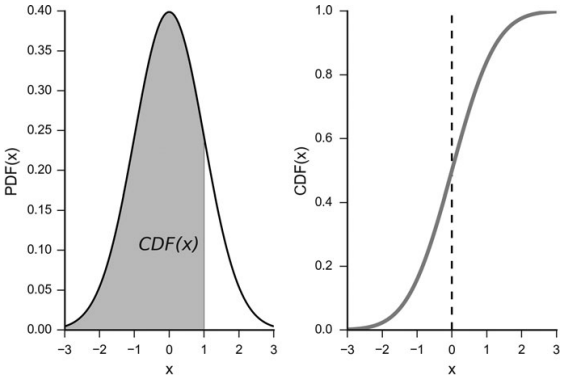

Fig. 6.1 Probability density function (*left*) and cumulative distribution function (*right*) of a normal distribution

The CDF is the integral of the PDF from minus infinity up to the given value (see Fig. 6.1), and thus specifies the percentage of the data that lie below this value. Knowing the CDF simplifies the calculation of how likely it is to find a value x within the range a to b (Fig. 5.2): The probability to find a value between a and b is given by the integral over the PDF in that range (see Fig. 5.2), and can be found by the difference of the corresponding CDF-values:

$$\mathbf{P}(a \leq X \leq b) = \int_a^b PDF(x)dx = CDF(b) - CDF(a) \tag{6.4}$$

For discrete distributions, the integral has to be replaced by a sum.

Coming back to *percentiles*: those are just the inverse of the CDF, and give the value below which a given percentage of the data values occur (see Fig. 6.5, lower left plot). While the expression "percentiles" does not come up very often, one will frequently encounter specific centiles:

- To find the range which includes 95 % of the data, one has to find the 2.5th and the 97.5th percentile of the sample distribution.
- The 50th percentile is the *median*.
- Also important are the *quartiles*, i.e., the 25th and the 75th percentile. The difference between them is called the *inter-quartile range* (*IQR*).

Median, upper, and lower quartile are used for the data display in box plots (Fig. 4.7).

c) Standard Deviation and Variance

Figure 5.1 shows a sketch of how a sample statistic relates to the corresponding population parameter. Applying this concept to the *variance* of a data set, we distinguish between the *sample variance*, i.e., the variance in the data sampled, and the *population variance*, i.e., the variance of the full population. The maximum likelihood estimator of the sample variance is given by

$$var = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n} \tag{6.5}$$

However, Eq. 6.5 systematically underestimates the population variance, and is therefore referred to as a "biased estimator" of the population variance. In other words, if you take a population with a given *population standard deviation*, and one thousand times selects n random samples from that population and calculate the standard deviation for each of these samples, then the mean of those one thousand *sample standard deviations* will be below the *population standard deviation*.

Figure 6.2 tries to motivate why the sample variance systematically underestimates the population variance: the sample mean is always chosen such that the variance of the given sample data is minimized, and thereby underestimates the variance of the population.

It can be shown that the best unbiased estimator for the population variance is given by

$$var = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1} \tag{6.6}$$

Equation 6.6 is referred to as *sample variance*.

The *standard deviation* is the square root of the variance, and the *sample standard deviation* the square root of the sample variance:

$$s = \sqrt{var} \tag{6.7}$$

As indicated in Table 5.1, in statistics it is common to denote the population standard deviation with σ, and the sample standard deviation with s.

Watch out: in contrast to other languages like *Matlab* or *R*, *numpy* by default calculates the variance for "n." To obtain the sample variance one has to set

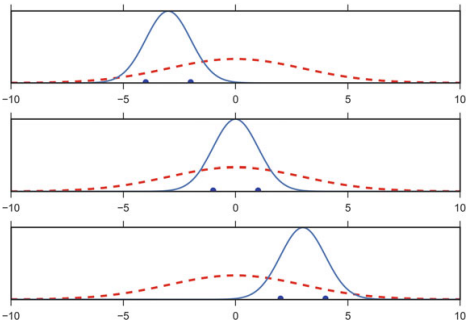

Fig. 6.2 Gaussian distributions fitted to selections of data from the underlying distribution: While the average mean of a number of samples converges to the real mean, the sample standard deviation underestimates the standard deviation from the distribution

"ddof=1":

```
In [1]: data = np.arange(7,14)

In [2]: np.std(data, ddof=0)
Out[2]: 2.0

In [3]: np.std(data, ddof=1)
Out[3]: 2.16025
```

Note: In *pandas*, the default for the calculation of the standard deviation is set to *ddof=1*.

d) Standard Error

The *standard error* is the estimate of the standard deviation of a coefficient. For example, in Fig. 6.3, we have 100 data points from a normal distribution about 5. The more data points we have to estimate the mean value, the better our estimate of the mean becomes.

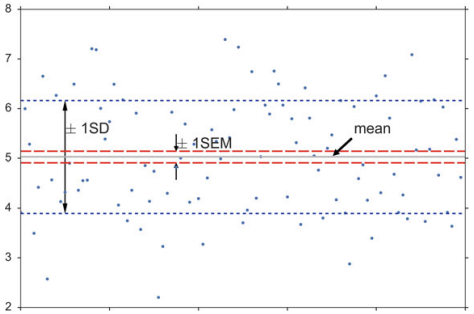

Fig. 6.3 One hundred random data points, from a normal distribution about 5. The sample mean (*solid line*) is very close to the real mean. The standard deviation of the mean (*long dashed line*), or standard error of the mean (SEM), is ten times smaller than the standard deviation of the samples (*short dashed line*)

For normally distributed data, the *sample standard error of the mean* (SE or SEM) is

$$SEM = \frac{s}{\sqrt{n}} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \cdot \frac{1}{\sqrt{n}} \tag{6.8}$$

So with 100 data points the standard deviation of our estimate, i.e., the standard error of the mean, is ten times smaller than the sample standard deviation.

e) Confidence Intervals

In the statistical analysis of data it is common to state the confidence interval of an estimated parameter. The $\alpha\%$ *confidence interval* (*CI*) reports the range that contains the true value for the parameter with a likelihood of $\alpha\%$.

If the sampling distribution is symmetrical and unimodal (i.e., decaying smoothly on both sides of the maximum), it will often be possible to approximate the confidence interval by

$$ci = mean \pm std * N_{PPF}\left(\frac{1-\alpha}{2}\right) \tag{6.9}$$

where *std* is the standard deviation, and N_{PPF} the *percentile point function* (*PPF*) for the standard normal distribution (see Fig.6.5). For the 95 % two-sided confidence intervals, for example, you have to calculate the *PPF(0.025)* of the standard normal distribution to get the lower and upper limit of the confidence interval. For a *Python* implementation for a normal distribution, see for example the code-sample on p. 106.

Notes

- To calculate the confidence interval for the mean value, the standard deviation has to be replaced by the standard error.
- If the distribution is skewed, Eq. 6.9 is *not* appropriate and does not provide the correct confidence intervals!

6.1.3 Parameters Describing the Form of a Distribution

In `scipy.stats`, continuous distribution functions are characterized by their *location* and their *scale*. To give two examples: for the normal distribution, (*location*/*shape*) are given by (*mean*/*standard deviation*) of the distribution; and for the uniform distribution, they are given by the (*start*/*end-start*) of the range where the distribution is different from zero.

a) Location

A *location parameter* x_0 determines the location or shift of a distribution:

$$f_{x0}(x) = f(x - x_0)$$

Examples of location parameters include the mean, the median, and the mode.

b) Scale

The *scale parameter* describes the width of a probability distribution. If the scale parameter s is large, then the distribution will be more spread out; if s is small then it will be more concentrated. If the probability density exists for all values of s, then the density (as a function of the scale parameter only) satisfies

$$f_s(x) = f(x/s)/s$$

where f is the density of a standardized version of the density.

c) Shape Parameters

It is customary to refer to all of the parameters beyond location and scale as *shape parameters*. Thankfully, almost all of the distributions that we use in statistics have only one or two parameters. It follows that the *skewness* and *kurtosis* of these distribution are constants.

Skewness

Distributions are *skewed* if they depart from symmetry (Fig. 6.4, left). For example, for measurements that cannot be negative, which is usually the case, we can infer that the data have a skewed distribution if the standard deviation is more than half the mean. Such an asymmetry is referred to as *positive skewness*. The opposite, negative skewness, is rare.

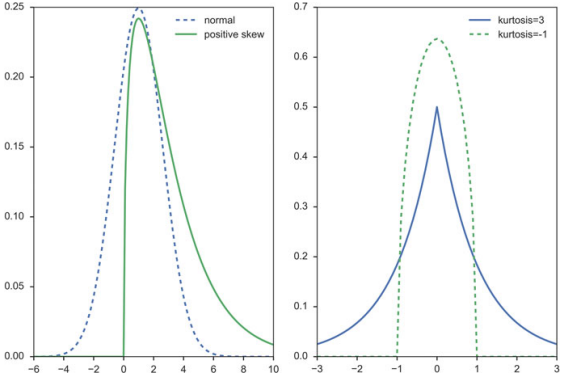

Fig. 6.4 (*Left*) Normal distribution, and distribution with positive skewness. (*Right*) The (leptokurtic) Laplace distribution has an excess kurtosis of 3, and the (platykurtic) Wigner semicircle distribution an excess kurtosis of −1

Kurtosis

Kurtosis is a measure of the "peakedness" of the probability distribution (Fig. 6.4, right). Since the normal distribution has a kurtosis of 3, the *excess kurtosis* = *kurtosis-3* is *0* for the normal distribution. Distributions with negative or positive excess kurtosis are called *platykurtic distributions* or *leptokurtic distributions*, respectively.

6.1.4 Important Presentations of Probability Densities

Figure 6.5 shows a number of functions that are equivalent to the PDF, but each represents a different aspect of the probability distribution. I will give examples which demonstrate each aspect for a normal distribution describing the size of male subjects.

- *Probability density function* (*PDF*): Note that to obtain the probability for the variable appearing in a certain interval, you have to integrate the PDF over that range.
  Example: What is the chance that a man is between 160 and 165 cm tall?

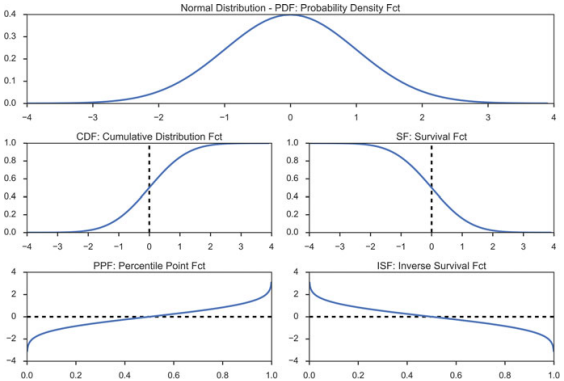

Fig. 6.5 Utility functions for continuous distributions, here for the standard normal distribution

- *Cumulative distribution function* (*CDF*): gives the probability of obtaining a value smaller than the given value.
  Example: What is the chance that a man is less than 165 cm tall?
- *Survival Function* (*SF*) $= 1 -$ CDF: gives the probability of obtaining a value larger than the given value. It can also be interpreted as the proportion of data "surviving" above a certain value.
  Example: What is the chance that a man is larger than 165 cm?
- *Percentile Point Function* (*PPF*): the inverse of the CDF. The PPF answers the question "Given a certain probability, what is the corresponding input value for the CDF?"
  Example: Given that I am looking for a man who is smaller than 95 % of all other men, what size does the subject have to be?
- *Inverse Survival Function* (*ISF*): the name says it all.
  Example: Given that I am looking for a man who is larger than 95 % of all other men, what size does the subject have to be?
- Random Variate Sample (*RVS*): random variates from a given distribution. (A *variable* is the general type, a *variate* is a specific number.)

Note: In *Python*, the most elegant way of working with distribution functions is a two-step procedure:

- In the first step, you create the distribution (e.g., `nd = stats.norm()`). Note that this is a distribution (in *Python* parlance a "frozen distribution"), not a function yet!
- In the second step, you decide which function you want to use from this distribution, and calculate the function value for the desired x-input (e.g., `y = nd.cdf(x)`).

```
import numpy as np
from scipy import stats

myDF = stats.norm(5,3)    # Create the frozen distribution

x = np.linspace(-5, 15, 101)
y = myDF.cdf(x)       # Calculate the corresponding CDF
```

6.2 Discrete Distributions

Two discrete distributions are frequently encountered: the *binomial distribution* and the *Poisson distribution*.

The big difference between those two functions: applications of the binomial distribution have an inherent upper limit (e.g., when you throw dice five times, each side can come up a maximum of five times); in contrast, the Poisson distribution does not have an inherent upper limit (e.g., how many people you know).

6.2.1 Bernoulli Distribution

The simplest case of a univariate distribution, and also the basis of the binomial distribution, is the *Bernoulli distribution* which has only two states, e.g., the simple coin flipping test. If we flip a coin (and the coin is not rigged), the chance that "heads" comes up is $p_{heads} = 0.5$. And since it has to be *heads* or *tails*, we must have

$$p_{heads} + p_{tails} = 1 \tag{6.10}$$

so the chance for "tails" is $p_{tails} = 1 - p_{heads}$.

We see that one parameter, $p = p_{heads}$, completely determines everything, and we can fix the distribution with the commands

```
In [1]: from scipy import stats
In [2]: p = 0.5
In [3]: bernoulliDist = stats.bernoulli(p)
```

In *Python* this is called a "frozen distribution function", and it allows us to calculate everything we want for this distribution. For example, the probability if head comes up zero or one times is given by the *probability mass function* (*PMF*)

```
In [4]: p_tails = bernoulliDist.pmf(0)
In [5]: p_heads = bernoulliDist.pmf(1)
```

And we can simulate 10 Bernoulli trials with

```
In [6]: trials = bernoulliDist.rvs(10)

In [7]: trials
Out[7]: array([0, 0, 0, 1, 0, 0, 0, 1, 1, 0])
```

In `In[6]`, *rvs* stands for *random variates*.

6.2.2 Binomial Distribution

If we flip a coin multiple times, and ask "How often did heads come up?" we have the *binomial distribution* (Fig. 6.6). In general, the binomial distribution is associated with the question "Out of a given (fixed) number of trials, how many will succeed?" Some example questions that are modeled with a binomial distribution are:

- Out of ten tosses, how many times will this coin land heads?
- From the children born in a given hospital on a given day, how many of them will be girls?
- How many students in a given classroom will have green eyes?
- How many mosquitoes, out of a swarm, will die when sprayed with insecticide?

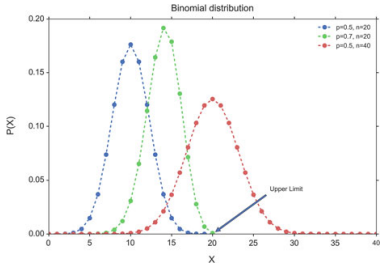

Fig. 6.6 Binomial distribution. Note that legal values exist only for integer x. The *dotted lines* in between only facilitate the grouping of the values to individual distribution parameters

We conduct n repeated experiments where the probability of success is given by the parameter p and add up the number of successes. This number of successes is represented by the random variable X. The value of X is then between 0 and n.

When a random variable X has a binomial distribution with parameters p and n we write it as $X \in B(n, p)$ and the probability mass function at $X = k$ is given by the equation:

$$P[X = k] = \begin{cases} \binom{n}{k} p^k (1-p)^{n-k} & 0 \le k \le n \\ 0 & \text{otherwise} \end{cases} \qquad 0 \le p \le 1, \quad n \in \mathbf{N} \tag{6.11}$$

$$P[X = k] = p^k (1-p)^{n-k} \qquad 0 \le p \le 1, \quad n \in \mathbf{N} \tag{6.12}$$

where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

In *Python*, the procedure is the same as above for the Bernoulli distribution, with one additional parameter, the number of coin tosses. First we generate the frozen distribution function, for example for four coin tosses:

```
In [1]: from scipy import stats
In [2]: import numpy as np

In [3]: (p, num) = (0.5, 4)
In [4]: binomDist = stats.binom(num, p)
```

and then we can calculate, e.g., the probabilities how often heads come up during those four tosses, given by the PMF for the values zero to four:

```
In [5]: binomDist.pmf(np.arange(5))
Out[5]: array([ 0.0625,  0.25  ,  0.375 ,  0.25  ,  0.0625])
```

For example, the chance that heads never comes up is about 6 %, the chance that it comes up exactly once is 25 %, etc.

Also note that the sum of all probabilities has to add up exactly to one:

$$p_0 + p_1 + \ldots + p_{n-1} = \sum_{i=0}^{n-1} p_i = 1 \tag{6.13}$$

b) Example: Binomial Test

Suppose we have a board game that depends on the roll of a die and attaches special importance to rolling a 6. In a particular game, the die is rolled 235 times, and 6 comes up 51 times. If the die is fair, we would expect 6 to come up 235/6 = 39.17 times. Is the proportion of 6's significantly higher than would be expected by chance, on the null hypothesis of a fair die?

To find an answer to this question using the *Binomial Test*, we consult the binomial distribution with $n = 235$ and $p = 1/6$, to determine the probability of finding exactly 51 sixes in a sample of 235 if the true probability of rolling a 6 on each trial is 1/6. We then find the probability of finding exactly 52, exactly 53, and so on up to 235, and add all these probabilities together. In this way, we calculate the probability of obtaining the observed result (51 sixes) or a more extreme result (> 51 sixes) assuming that the die is fair. In this example, the result is 0.0265, which indicates that observing 51 sixes is unlikely (not significant at the 5 % level) to come from a die that is not loaded to give many sixes (one-tailed test).

Clearly a die could roll too few sixes as easily as too many and we would be just as suspicious, so we should use the two-tailed test which (for example) splits the 5 % probability across the two tails. Note that to do this we cannot simply double the one-tailed p-value unless the probability of the event is 1/2. This is because the binomial distribution becomes asymmetric as that probability deviates from 1/2. "scipy.stats" therefore provides for the two-sided test the function "binom_test". (See also the explanation of one- and two-tailed *t*-tests, p. 141.)

Code: "ISP_binomial.py"[1]: Example of a one- and two-sided binomial test.

Table 6.1 Properties of discrete distributions

	Mean	Variance
Binomial	$n \cdot p$	$np(1-p)$
Poisson	λ	λ

[1] https://github.com/thomas-haslwanter/statsintro_python/tree/master/ISP/Code_Quantlets/06_Distributions/binomialTest.

6.2.3 Poisson Distribution

Any French speaker will notice that "Poisson" means "fish," but really there's nothing fishy about this distribution. It's actually pretty straightforward. The name comes from the mathematician Siméon-Denis Poisson (1781–1840).

The Poisson distribution is very similar to the binomial distribution. We are examining the number of times an event happens. The difference is subtle. Whereas the binomial distribution looks at how many times we register a success over a fixed total number of trials, the Poisson distribution measures how many times a discrete event occurs, over a period of continuous space or time. There is no "total" value n, and the Poisson distribution is defined by a single parameter.

The following questions can be answered with the Poisson distribution:

- How many pennies will I encounter on my walk home?
- How many children will be delivered at the hospital today?
- How many products will I sell after airing a new television commercial?
- How many mosquito bites did you get today after having sprayed with insecticide?
- How many defects will there be per 100 m of rope sold?

What's a little different about this distribution is that the random variable X which counts the number of events can take on any nonnegative integer value. In other words, I could walk home and find no pennies on the street. I could also find one penny. It's also possible (although unlikely, short of an armored-car exploding nearby) that I would find 10 or 100 or 10,000 pennies.

Instead of having a parameter p that represents a component probability as in the binomial distribution, this time we have the parameter "lambda" or λ which represents the "average or expected" number of events to happen within our experiment (Fig. 6.7). The probability mass function of the Poisson distribution is

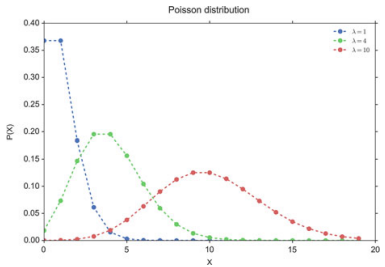

Fig. 6.7 Poisson distribution. Again note that legal values exist only for integer x. The *dotted lines* in between only facilitate the grouping of the values to individual distribution parameters

given by

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!} \tag{6.14}$$

.

Code: "ISP_distDiscrete.py"[2] shows different discrete distribution functions.

6.3 Normal Distribution

The Normal distribution or Gaussian distribution is by far the most important of all the distribution functions (Fig. 6.8). This is due to the fact that the mean values of *all* distribution functions approximate a normal distribution for large enough sample numbers (see Sect. 6.3.2). Mathematically, the normal distribution is characterized by a mean value μ, and a standard deviation σ:

$$f_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{6.15}$$

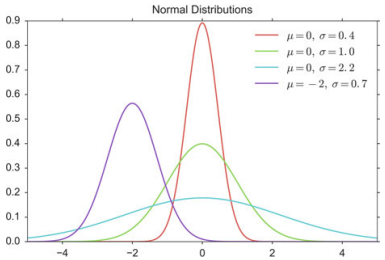

Fig. 6.8 Normal distributions, with different parameters for μ and σ

[2] https://github.com/thomas-haslwanter/statsintro_python/tree/master/ISP/Code_Quantlets/06_Distributions/distDiscrete.

where $-\infty < x < \infty$, and $f_{\mu,\sigma}$ is the *Probability Density Function* (*PDF*) of the normal distribution. In contrast to the PMF (probability mass function) of discrete distributions, which is defined only for discrete integers, the PDF is defined for continuous values. The *standard normal distribution* is a normal distribution with a mean of zero and a standard deviation of one, and is sometimes referred to as *z-distribution*.

For smaller sample numbers, the sample distribution can show quite a bit of variability. For example, look at 25 distributions generated by sampling 100 numbers from a normal distribution (Fig. 6.9)

The normal distribution with parameters μ and σ is denoted as $N(\mu, \sigma)$ (Table 6.2). If the random variates (rvs) of X are normally distributed with expectation μ and standard deviation σ, one writes: $X \in N(\mu, \sigma)$ (Fig. 6.10).

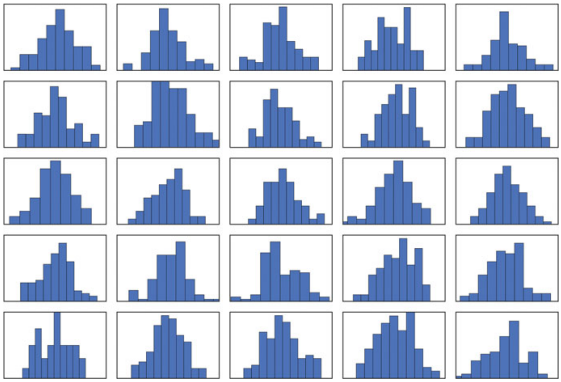

Fig. 6.9 Twenty-five randomly generated samples of 100 points from a standard normal distribution

Table 6.2 Tails of a normal distribution, with the distance from the mean expressed in standard deviations (SDs)

	Probability of being	
Range	Within range	Outside range
Mean ± 1SD	68.3 %	31.7 %
Mean ± 2SD	95.4 %	4.6 %
Mean ± 3SD	99.7 %	0.27 %

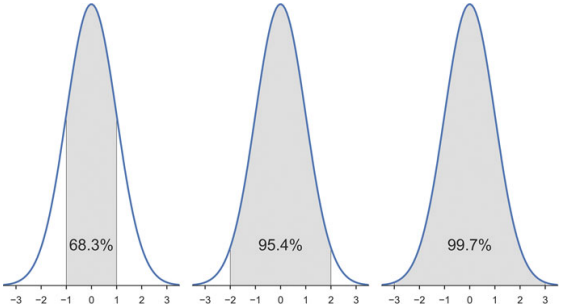

Fig. 6.10 Area under ± 1, 2, and 3 standard deviations of a normal distribution

Code: "ISP_distNormal.py"[3] shows simple manipulations of normal distribution functions.

```
In [1]:  import numpy as np
In [2]:  from scipy import stats

In [3]:  mu = -2
In [4]:  sigma = 0.7
In [5]:  myDistribution = stats.norm(mu, sigma)
In [6]:  significanceLevel = 0.05

In [7]:  myDistribution.ppf(
             [significanceLevel/2, 1-significanceLevel/2] )
Out[8]:  array([-3.38590382, -0.61409618]
```

Example of how to calculate the interval of the PDF containing 95 % of the data, for the blue curve in Fig. 6.8

[3] https://github.com/thomas-haslwanter/statsintro_python/tree/master/ISP/Code_Quantlets/06_Distributions/distNormal.

Sum of Normal Distributions
An important property of normal distributions is that the sum (or difference) of two normal distributions is also normally distributed. i.e., if

$$X \in N(\mu_X, \sigma_X^2)$$
$$Y \in N(\mu_Y, \sigma_Y^2)$$
$$Z = X \pm Y,$$

then

$$Z \in N(\mu_X \pm \mu_Y, \sigma_X^2 + \sigma_Y^2). \tag{6.16}$$

In words, the variance of the sum is the sum of the variances.

6.3.1 Examples of Normal Distributions

- If the average man is 175 cm tall with a standard deviation of 6 cm, what is the probability that a man selected at random will be 183 cm tall?
- If cans are assumed to have a standard deviation of 4 g, what does the average weight need to be in order to ensure that 99 % of all cans have a weight of at least 250 g?
- If the average man is 175 cm tall with a standard deviation of 6 cm, and the average woman is 168 cm tall with a standard deviation of 3 cm, what is the probability that a randomly selected man will be shorter than a randomly selected woman?

6.3.2 Central Limit Theorem

The central limit theorem states that the mean of a sufficiently large number of identically distributed random variates will be approximately normally distributed. Or in other words, the sampling distribution of the mean tends toward normality, regardless of the distribution. Figure 6.11 shows that averaging over ten uniformly distributed data already produces a smooth, almost Gaussian distribution.

Code: "ISP_centralLimitTheorem.py"[4] demonstrates that already averaging over ten uniformly distributed data points produces an almost Gaussian distribution.

[4] https://github.com/thomas-haslwanter/statsintro_python/tree/master/ISP/Code_Quantlets/06_Distributions/centralLimitTheorem.

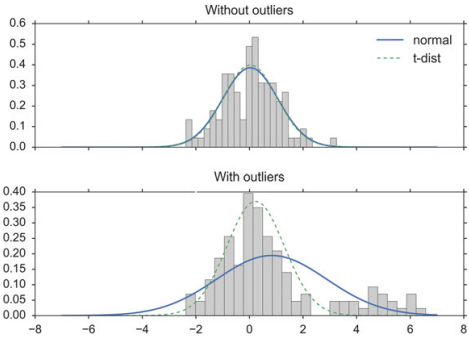

Fig. 6.11 Demonstration of the *Central Limit Theorem* for a uniform distribution: (*Left*) Histogram of uniformly distributed random data between 0 and 1. (*Center*) Histogram of average over two data points. (*Right*) Histogram of average over ten data points

6.3.3 Distributions and Hypothesis Tests

To illustrate the connection between distribution functions and hypothesis tests, let me go step-by-step through the analysis of the following problem:

The average weight of a newborn child in the USA is 3.5 kg, with a standard deviation of 0.76 kg. If we want to check all children that are significantly different from the typical baby, what should we do with a child that is born with a weight of 2.6 kg?

We can rephrase that problem in the form of a *hypothesis test*: our hypothesis is that *the baby comes from the population of healthy babies*. Can we keep the hypothesis, or does the weight of the baby suggest that we should reject that hypothesis?

To answer that question, we can proceed as follows:

- Find the distribution that characterizes healthy babies $\rightarrow \mu = 3.5$, $\sigma = 0.76$.
- Calculate the CDF at the value of interest $\rightarrow CDF(2.6\,kg) = 0.118$. In other words, the probability that a healthy baby is at least 0.9 kg lighter than the average baby is 11.8 %.
- Since we have a normal distribution, the probability that a healthy baby is at least 0.9 kg heavier than the average baby is also 11.8 %.

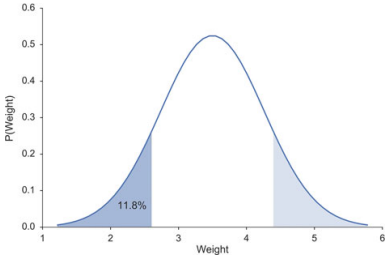

Fig. 6.12 The chance that a healthy baby weighs 2.6 kg or less is 11.8 % (*darker area*). The chance that the difference from the mean is as extreme or more extreme than for 2.6 kg is twice that much, as the lighter area must also be considered

- Interpret the result → *If the baby is healthy, the chance that its weight deviates by at least 0.9 kg from the mean is 2*11.8 % = 23.6 %. This is not significant, so we do not have sufficient evidence to reject our hypothesis, and our baby is regarded as healthy* (see Fig. 6.12).

```
In [1]: from scipy import stats

In [2]: nd = stats.norm(3.5, 0.76)

In [3]: nd.cdf(2.6)
Out[3]: 0.11816
```

Note: The starting hypothesis is often referred to as *null hypothesis*. In our example it would mean that we assume that there is *null* difference between the distribution the baby comes from and the population of healthy babies.

6.4 Continuous Distributions Derived from the Normal Distribution

Some frequently encountered continuous distributions are closely related to the normal distribution:

- ***t*-Distribution**: The sample distribution of mean values for samples from a normally distributed population. Typically used for small sample numbers, when the true mean/SD are not known.
- **χ-Square distribution**: For describing variability of normally distributed data.

- ***F*-distribution**: For comparing variabilities of two sets of normally distributed data.

In the following we will discuss these continuous distribution functions.

Code: "ISP_distContinuous.py"[5] shows different continuous distribution functions.

6.4.1 t-Distribution

In 1908 W.S. Gosset, who worked for the Guinness brewery in Dublin, was interested in the problems of small samples, for example the chemical properties of barley where sample sizes might be as low as 3. Since in these measurements the true variance of the mean was unknown, it must be approximated by the sample standard error of the mean. And the ratio between the sample mean and the standard error had a distribution that was unknown till Gosset, under the pseudonym "Student," solved that problem. The corresponding distribution is the *t-distribution*, and converges for larger values towards the normal distribution (Fig. 6.13). Due to Gosset's pseudonym, "Student," it is now also known as *Student's distribution*.

Since in most cases the population mean and its variance are unknown, one typically works with the *t*-distribution when analyzing sample data.

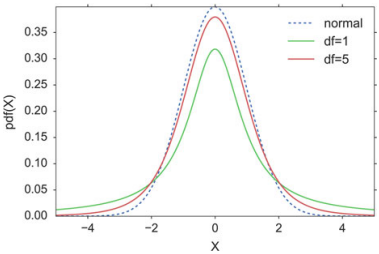

Fig. 6.13 *t*-Distribution

[5] https://github.com/thomas-haslwanter/statsintro_python/tree/master/ISP/Code_Quantlets/06_Distributions/distContinuous.

If $\bar{x}$ is the sample mean, and s the sample standard deviation, the resulting statistic is

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}} = \frac{\bar{x} - \mu}{SE} \tag{6.17}$$

A very frequent application of the *t*-distribution is in the calculation of confidence intervals for the mean. The width of the 95 %-confidence interval (CI), i.e., the interval that contains the true mean with a chance of 95 %, is the same width about the population mean that contains 95 % of the sample means (Eq. 6.17):

$$ci = mean \pm se * t_{df,\alpha} \tag{6.18}$$

The following example shows how to calculate the *t*-values for the 95 %-CI, for n = 20. The lower end of the 95 % CI is the value that is larger than 2.5 % of the distribution; and the upper end of the 95 %-CI is the value that is larger than 97.5 % of the distribution. These values can be obtained either with the *percentile point function* (*PPF*), or with the *inverse survival function* (*ISF*). For comparison, I also calculate the corresponding value from the normal distribution:

```
In [1]: import numpy as np
In [2]: from scipy import stats
In [3]: n = 20
In [4]: df = n-1
In [5]: alpha = 0.05

In [6]: stats.t(df).isf(alpha/2)
Out[6]: 2.093

In [7]: stats.norm.isf(alpha/2)
Out[7]: 1.960
```

In *Python*, the 95 %-CI for the mean can be obtained with a one-liner:

```
In [8]: alpha = 0.95
In [9]: df = len(data)-1
In [10]: ci = stats.t.interval(alpha, df,
                 loc=np.mean(data), scale=stats.sem(data))
```

Since the *t*-distribution has longer tails than the normal distribution, it is much less affected by extreme cases (see Fig. 6.14).

6.4.2 Chi-Square Distribution

a) Definition

The chi-square distribution is related to the normal distribution in a simple way: if a random variable X has a normal distribution ($X \in N(0,1)$), then X^2 has a chi-square distribution, with one degree of freedom ($X^2 \in \chi^2_1$). The sum squares of n

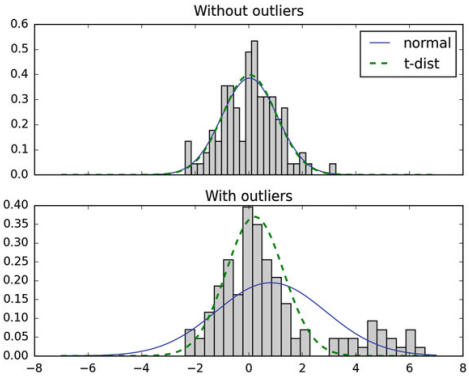

Fig. 6.14 The *t*-distribution is much more robust against outliers than the normal distribution. (*Top*) Best-fit normal and *t*-distribution, for a sample from a normal population. (*Bottom*) Same fits, with 20 "outliers," normally distributed data about 5, added

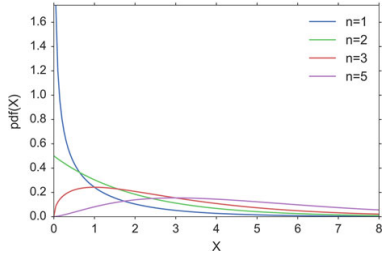

Fig. 6.15 Chi-square distribution

independent and standard normal random variables have a chi-square distribution with n degrees of freedom (Fig. 6.15):

$$\sum_{i=1}^{n} X_i^2 \in \chi_n^2 \tag{6.19}$$

b) Application Example

A pill producer is ordered to deliver pills with a standard deviation of $\sigma = 0.05$. From the next batch of pills $n = 13$ random samples have a weight of 3.04, 2.94, 3.01, 3.00, 2.94, 2.91, 3.02, 3.04, 3.09, 2.95, 2.99, 3.10, 3.02 g.

Question Is the standard deviation larger than allowed?

Answer Since the chi-square distribution describes the distribution of the summed squares of random variates from a *standard normal distribution*, we have to normalize our data before we calculate the corresponding CDF-value:

$$SF_{\chi^2_{(n-1)}} = 1 - CDF_{\chi^2_{(n-1)}} \left(\sum (\frac{x - \bar{x}}{\sigma})^2 \right) = 0.1929 \tag{6.20}$$

Interpretation If the batch of pills is from a distribution with a standard deviation of $\sigma = 0.05$, the likelihood of obtaining a chi-square value as large or larger than the one observed is about 19 %, so it is not atypical. In other words, the batch matches the expected standard deviation.

Note The number of the DOF is $n - 1$, because we are only interested in the shape of the distribution, and the mean value of the n data is subtracted from all data points.

```
In [1]: import numpy as np
In [2]: from scipy import stats
In [3]: data = np.r_[3.04, 2.94, 3.01, 3.00, 2.94, 2.91, 3.02,
                     3.04, 3.09, 2.95, 2.99, 3.10, 3.02]
In [4]: sigma = 0.05
In [5]: chi2Dist = stats.chi2(len(data)-1)
In [6]: statistic = sum( ((data-np.mean(data))/sigma)**2 )

In [7]: chi2Dist.sf(statistic)
Out[7]: 0.19293
```

6.4.3 F-Distribution

a) Definition

This distribution is named after Sir Ronald Fisher, who developed the *F* distribution for use in determining critical values in ANOVAs ("ANalysis Of VAriance," see Sect. 8.3.1).

If we want to investigate whether two groups have the same variance, we have to calculate the ratio of the sample standard deviations squared:

$$F = \frac{S_x^2}{S_y^2} \tag{6.21}$$

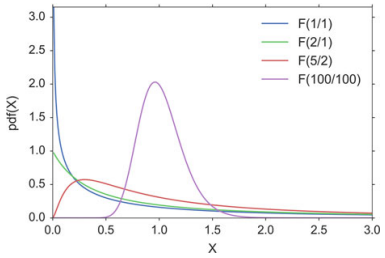

Fig. 6.16 *F*-distribution

where S_x is the sample standard deviation of the first sample, and S_y the sample standard deviation of the second sample.

The distribution of this statistic is the *F distribution*. For applications in ANOVAs (see Sect. 8.3.1), the cutoff values for an F distribution are generally found using three variables:

- ANOVA numerator degrees of freedom
- ANOVA denominator degrees of freedom
- significance level

An ANOVA compares the size of the variance between two different samples. This is done by dividing the larger variance by the smaller variance. The formula for the resulting F statistic is (Fig. 6.16):

$$F(r_1, r_2) = \frac{\chi^2_{r1}/r_1}{\chi^2_{r2}/r_2} \tag{6.22}$$

where χ^2_{r1} and χ^2_{r2} are the chi-square statistics of sample one and two respectively, and r_1 and r_2 are their degrees of freedom.

b) Application Example

Take for example the case where we want to compare the precision of two methods to measure eye movements. The two methods can have different accuracy and different precision. As shown in Fig. 6.17, the *accuracy* gives the deviation between the real and the measured value, while the *precision* is determined by the variance of the measurements. With the test we want to determine if the precision of the two methods is equivalent, or if one method is more precise than the other.

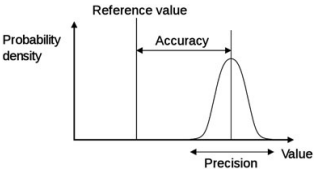

Fig. 6.17 Accuracy and precision of a measurement are two different characteristics

When you look 20° to the right, you get the following results:

Method 1: [20.7, 20.3, 20.3, 20.3, 20.7, 19.9, 19.9, 19.9, 20.3, 20.3, 19.7, 20.3]
Method 2: [19.7, 19.4, 20.1, 18.6, 18.8, 20.2, 18.7, 19.]

The F statistic is $F = 0.244$, and has $n-1$ and $m-1$ degrees of freedom, where n and m are the number of recordings with each method. The code sample below shows that the F statistic is in the tail of the distribution (p_oneTail=0.019), so we reject the hypothesis that the two methods have the same precision.

```
import numpy as np
from scipy import stats

method1 = np.array([20.7,  20.3,  20.3,  20.3,  20.7,  19.9,
                    19.9, 19.9,  20.3, 20.3,  19.7,  20.3])
method2 = np.array([ 19.7,  19.4,  20.1,  18.6,  18.8,  20.2,
                     18.7,  19. ])

fval = np.var(method1, ddof=1)/np.var(method2, ddof=1)
fd = stats.f(len(method1)-1,len(method2)-1)
p_oneTail = fd.cdf(fval)    # -> 0.019

if (p_oneTail<0.025) or (p_oneTail>0.975):
    print('There is a significant difference'
          ' between the two distributions.')
else:
    print('No significant difference.')
```

6.5 Other Continuous Distributions

Some common distributions which are not directly related to the normal distribution are described briefly in the following:

- **Lognormal distribution**: A normal distribution, plotted on an exponential scale. A logarithmic transformation of the data is often used to convert a strongly skewed distribution into a normal one.
- **Weibull distribution**: Mainly used for reliability or survival data.

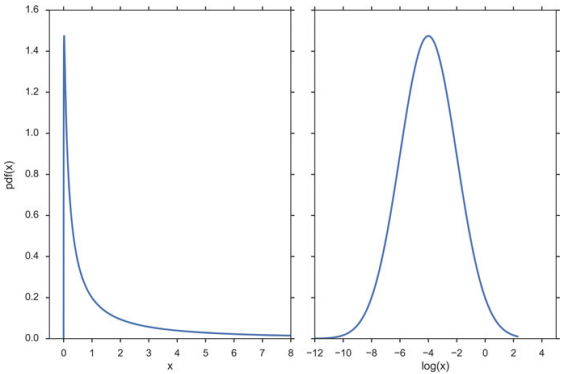

Fig. 6.18 Lognormal distribution, plotted against a linear abscissa (*left*) and against a logarithmic abscissa (*right*)

- **Exponential distribution**: Exponential curves.
- **Uniform distribution**: When everything is equally likely.

6.5.1 Lognormal Distribution

Normal distributions are the easiest ones to work with. In some circumstances a set of data with a positively skewed distribution can be transformed into a symmetric, normal distribution by taking logarithms. Taking logs of data with a skewed distribution will often give a distribution that is near to normal (see Fig. 6.18).

6.5.2 Weibull Distribution

The Weibull distribution is the most commonly used distribution for modeling reliability data or "survival" data. It has two parameters, which allow it to handle increasing, decreasing, or constant failure-rates (see Fig. 6.19). It is defined as

$$f_x(x) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0, \\ 0 & x < 0, \end{cases} \tag{6.23}$$

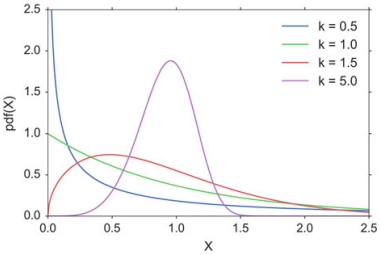

Fig. 6.19 Weibull distribution. The scale parameters for all curves is $\lambda = 1$

where $k > 0$ is the *shape parameter* and $\lambda > 0$ is the scale parameter of the distribution. (It is one of the rare cases where we use a shape parameter different from skewness and kurtosis.) Its complementary cumulative distribution function is a stretched exponential function.

If the quantity x is a "time-to-failure," the Weibull distribution gives a distribution for which the failure rate is proportional to a power of time. The shape parameter, k, is that power plus one, and so this parameter can be interpreted directly as follows:

- A value of $k < 1$ indicates that the failure rate decreases over time. This happens if there is significant "infant mortality," or defective items failing early and the failure rate decreasing over time as the defective items are weeded out of the population.
- A value of $k = 1$ indicates that the failure rate is constant over time. This might suggest random external events are causing mortality, or failure.
- A value of $k > 1$ indicates that the failure rate increases with time. This happens if there is an "aging" process, or parts that are more likely to fail as time goes on. An example would be products with a built-in weakness that fail soon after the warranty expires.

In the field of materials science, the shape parameter k of a distribution of strengths is known as the *Weibull modulus*.

6.5.3 Exponential Distribution

For a stochastic variable X with an exponential distribution, the probability distribution function is:

$$f_x(x) = \begin{cases} \lambda e^{-\lambda x}, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{6.24}$$

The exponential PDF is shown in Fig. 6.20.

6.5.4 Uniform Distribution

This is a simple one: an even probability for all data values (Fig. 6.21). Not very common for real data.

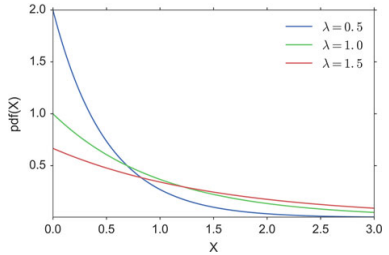

Fig. 6.20 Exponential distribution

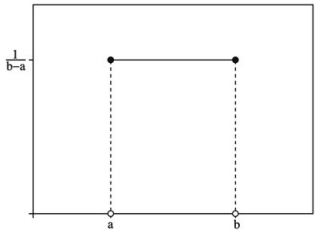

Fig. 6.21 Uniform distribution

6.6 Exercises

6.1 Sample Standard Deviation

Create an numpy-array, containing the data $1, 2, 3, \ldots, 10$. Calculate mean and sample(!)-standard deviation. (Correct answer for the SD: 3.03.)

6.2 Normal Distribution

- Generate and plot the Probability Density Function (PDF) of a normal distribution, with a mean of 5 and a standard deviation of 3.
- Generate 1000 random data from this distribution.
- Calculate the standard error of the mean of these data. (Correct answer: ca. 0.096.)
- Plot the histogram of these data.
- From the PDF, calculate the interval containing 95 % of these data. (Correct answer: $[-0.88, 10.88]$.)
- Your doctor tells you that he can use hip implants for surgery even if they are 1 mm bigger or smaller than the specified size. And your financial officer tells you that you can discard 1 out of 1000 hip implants, and still make a profit.

  What is the required standard deviation for the producer of the hip implants, to simultaneously satisfy both requirements? (Correct answer: $\sigma = 0.304$ mm.)

6.3 Continuous Distributions

- ***t*-Distribution:** Measuring the weight of your colleagues, you have obtained the following weights: 52, 70, 65, 85, 62, 83, 59 kg. Calculate the corresponding mean, and the 99 % confidence interval for the mean. Note: with n values you have $n - 1$ DOF for the *t*-distribution. (Correct answer: $68.0 + / - 17.2$ kg.)
- **Chi-square Distribution:** Create three normally distributed data sets (mean $=$ 0, SD $= 1$), with 1000 samples each. Then square them, sum them (so that you have 1000 data-points), and create a histogram with 100 bins. This should be similar to the curve for the chi-square distribution, with 3 DOF (i.e., it should come down at the left, see Fig. 6.22).
- ***F*-Distribution:** You have two apple trees. There are three apples from the first tree that weigh 110, 121, and 143 g, respectively, and four from the other which weigh 88, 93, 105, and 124 g, respectively. Are the variances from the two trees different?

  Note: calculate the corresponding F-value, and check if the CDF for the corresponding F-distribution is < 0.025. (Correct answer: no.)

6.4 Discrete Distributions

- **Binomial Distribution**: "According to research, pure blue eyes in Europe approach greatest frequency in Finland, Sweden, and Norway (at 72 %), followed by Estonia, Denmark (69 %); Latvia, Ireland (66 %); Scotland (63 %); Lithuania (61 %); The Netherlands (58 %); Belarus, England (55 %); Germany (53 %); Poland, Wales (50 %); Russia, The Czech Republic (48 %); Slovakia (46 %);

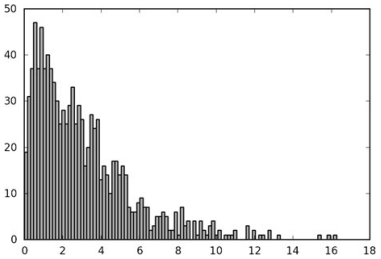

Fig. 6.22 Chi2 distribution with three degrees of freedom

Belgium (43 %); Austria, Switzerland, Ukraine (37 %); France, Slovenia (34 %); Hungary (28 %); Croatia (26 %); Bosnia and Herzegovina (24 %); Romania (20 %); Italy (18 %); Serbia, Bulgaria (17 %); Spain (15 %); Georgia, Portugal (13 %); Albania (11 %); Turkey and Greece (10 %). Further analysis shows that the average occurrence of blue eyes in Europe is 34 %, with 50 % in Northern Europe and 18 % in Southern Europe."

If we have 15 Austrian students in the classroom, what is the chance of finding three, six, or ten students with blue eyes? (Correct answer: 9, 20.1, and 1.4 %.)

- **Poisson Distribution**: In 2012 there were 62 fatal accidents on streets in Austria. Assuming that those are evenly distributed, we have on average $62/(365/7) = 1.19$ fatal accidents per week. How big is the chance that in a given week there are no, two, or five accidents? (Correct answer: 30.5, 21.5, 0.6 %.)