# Chapter 7: Normalizing and Redistributing Variables

## Overview

From this point on in preparing the data, all of the variables in a data set have a numerical representation. Chapter 6 explained why and how to find a suitable and appropriate numerical representation for alpha values—that is, the one that either reveals the most information, or at least does the least damage to existing information. The only time that an alpha variable's label values come again to the fore is in the Prepared Information Environment Output module, when the numerical representations of alpha values have to be remapped into the appropriate alpha representation. The discussion in most of the rest of the book assumes that the variables not only have numerical values, but are also normalized across the range of 0–1. Why and how to normalize the range of a variable is covered in the first part of this chapter.

In addition to looking at the range of a variable, its distribution may also make problems. The way a variable's values are spread, or distributed, across its range is known as its *distribution*. Some patterns in a variable's distribution can cause problems for modeling tools. These patterns may make it hard or impossible for the modeling tool to fully access and use the information a variable contains. The second topic in this chapter looks at normalizing the distribution, which is a way to manipulate a variable's values to alleviate some of these problems.

The chapter, then, covers two key topics: normalizing the range of a variable and normalizing the distribution of a variable. (Neither of these normalization methods have anything in common with putting data into the multitable structures called "normal form" in a database, data warehouse, or other data repository.) During the process of manipulation, as well as exposing information, there is useful insight to be gained about the nature of the variables and the data. Some of the potential insights are briefly discussed in this chapter, although the full exploration of these relationships properly forms part of the data survey.

## 7.1 Normalizing a Variable's Range

Chapter 6, discussing state space, pointed out that it was convenient to normalize variable ranges across the span of 0–1. Convenience is not an attribute to be taken lightly. Using anything less than the most convenient methods hardly contributes to easy and efficient completion of a task. However, some modeling tools *require* the range of the input to be normalized. For example, the neurons in most neural-network-based tools require data to be close to the range of 0 to 1, or –1 to +1, depending on the type of neuron. (More on neural networks in Chapter 10.) Most tools that do not actually require

range normalization may benefit from it, sometimes enormously. (Chapter 2 mentioned, for instance, that exposing information and easing the learning task can reduce an effect known as *feature swamping*.)

Normalization methods represent compromises designed to achieve particular ends. Normalization requires taking values that span one range and representing them in another range. This requires remapping values from an input range to an output range. Each method of remapping may introduce various distortions or biases into the data. Some biases and distortions are deliberately introduced to better expose information content. Others are unknowingly or accidentally introduced, and damage information exposure. Some types of bias and distortion introduced in some normalization processes are beneficial only for particular types of data, or for particular modeling methods. Automated data preparation must use a method that is generally applicable to any variable range and type—one that at least does no harm to the information content of the variable. Ideally, of course, the normalization method should be beneficial.

Any method of addressing the problems has its own trade-offs and introduces biases and distortions that must be understood. Some commercial tools normalize variables. When they do, it can cause a problem if the tool uses a default method that the modeler cannot control. Exactly what might be lost in the normalization, or what distortion might be introduced, is hard to know if the normalization method is not in the modeler's control, or worse, not even known to the modeler. (The neural network model comparison between prepared data and "unprepared" data in Chapter 12 in part demonstrates this issue.)

Methods of normalization are plentiful. Some do more than one thing at a time. They not only normalize ranges, but also address various problems in the distribution of a variable. The data preparation process, as described in this book, deals with distribution problems as a separate issue (discussed later in this chapter), so normalization methods that adjust and correct simultaneously for range and distribution problems are not used. As far as range normalization goes, what the modeler needs is a method that normalizes the range of a variable, introducing as little distortion as possible, and is tolerant of out-of-range values.

Range normalization addresses a problem with a variable's range that arises because the data used in data preparation is necessarily only a sample of the population. (Chapter 5 discussed sampling.) Because a sample is used, there is a less than 100% confidence that the sample is fully representative of the population. This implies, among other things, that there is a less than 100% confidence that the maximum and minimum values of the range of a variable have been discovered. This in turn implies, with some degree of confidence, that values larger than the sample maximum, or smaller than the sample minimum, will turn up in the population—and more importantly, in other samples of the population. Since values that are outside the limits discovered in a sample are out of the range of the sample, they are called here *out-of-range* values. This only indicates that such values are out of the range discovered in the sample used for data preparation. They

certainly aren't out of the range of the population, only out of the range established in a particular sample—the training sample. Dealing with these out-of-range values presents a problem that has to be addressed. We need to look at what these problems are before considering how to fix them.

What problems turn up with out-of-range values? The answer to this question depends on the stage in the data exploration process in which the out-of-range value is encountered. The stages in which a problem might occur are during modeling: the training stage, the testing stage, and the execution stage. Preparation and survey won't come across out-of-range values as they work with the same sample. The modeling phase might have problems with out-of-range values, and a brief review of modeling stages will provide a framework to understand what problems the out-of-range values cause in each stage.

### 7.1.1 Review of Data Preparation and Modeling (Training, Testing, and Execution)

Chapter 3 described the creation, use, and purpose of the PIE, which is created during data preparation. It has two components: the PIE-Input component (PIE-I) that dynamically takes a training-input or live-input data set and transforms it for use by the modeling tool, and the PIE-Output component (PIE-O) that takes the output (predictions) from a model and transforms it back into "real-world" values. A representative sample of data is required to build the PIE. However, while this representative sample *might* be the one also used to build the (predictive, inferential, etc.) model, that is not necessarily so. The modeler may choose to use a different data set for modeling, from the one used to build the PIE. Creating the model requires at least training and testing phases, followed by execution when the model is applied to "live" data.

This means that there are potentially any number of sample data sets. During training, there is one data set for building the PIE, one (probably the same one) for building a model, and one (definitely a separate one) for testing the model. At execution time, any number of data sets may be run through the PIE-I, the model, and the PIE-O in order, say, to make predictions. For example, in a transaction system scoring individual transactions for some feature, say, fraud, each transaction counts as an input execution data set. Each transaction is separately presented to the PIE-I, to the scoring model, the results to the PIE-O, with the individual output score being finally evaluated, either manually or automatically. The transactions are not prepared as a batch in advance for modeling all together, but are individually presented for evaluation as they come in.

When building the PIE, it is easy to discover the maximum and minimum values in the sample data set. So no out-of-range values can occur when building the PIE. With any other sample data set, it is always possible to encounter an out-of-range value. Since the PIE provides the modeling environment, it is the PIE that must deal with the problems.

### 7.1.2 The Nature and Scope of the Out-of-Range Values

## Problem

Since the PIE knows the maximum and minimum values of the data sample, no out-of-range value can occur at this stage during its construction. However, what the modeler should ask is, What can I learn about the out-of-range values that are *expected* to occur in the population? The PIE is going to have to deal with out-of-range numbers when they turn up, so it needs to know the expected largest and smallest numbers it will encounter during execution. It is also useful to know how often an out-of-range number is likely to be found in the population.

There are two problems with out-of-range numbers. First, the PIE is not going to have any examples of these values, so it needs to estimate their range and frequency to determine suitable adjustments that allow for them. They are certain to turn up in the population, and the PIE will have to deal with them in some way that best preserves the information environment surrounding the model. The second problem is that the out-of-range values represent part of the information pattern in the population that the modeling tool is not going to be exposed to during training. The model can't see them during training because they aren't in the training sample. The modeler needs an estimate of the range and the proportion of values in the population that are not represented in the sample. This estimate is needed to help determine the model's range of applicability and robustness when it is exposed to real-world data. Clearly, the model cannot be expected to perform well on patterns that exist in the population when they are not modeled since they aren't in the training sample. The extent and prevalence of such patterns need to be as clearly delimited as possible.

Of course, the modeler, together with the domain expert and problem owner, will try to choose a level of confidence for selecting the sample that limits the problem to an acceptable degree. However, until a sample is taken, and the actual distribution of each variable sampled and profiled, the exact extent of the problem cannot be assessed. In any case, limiting the problem by setting confidence limits assumes that sufficient data is available to meet the confidence criteria chosen. When the training data set is limited in size, it may well be the amount of data available that is the limiting factor. In which case, the modeler needs to know the limits set by the data available. Unless the population is available for modeling, this is a problem that simply cannot be avoided.

The information about the model limits due to out-of-range values, although generated when creating the PIE modules, is generally reported as part of the data survey. It is important to note that although the information enfolded in the data in the out-of-range values is properly part of the population, the model will experience the previously unseen values as noise. Chapter 11 looks briefly at noise maps. A full survey assesses, where possible, how much noise comes from each measurable source, including out-of-range values. Unfortunately, space limitations preclude further discussion of methods for assessing noise contribution due to out-of-range values, and for separating it from noise from other sources.

### 7.1.3 Discovering the Range of Values When Building the PIE

How, then, does the miner determine the range and the frequency of values present in the population, but not in the sample? Recall that the data sample was determined to represent the population with a specific level of confidence. That confidence level is almost always less than 100%. A 95% confidence means that there remains a 5% confidence—that is, 1 in 20—that the sample is not representative. It doesn't need detailed analysis to see that if the range has been captured to a 95% confidence limit, out-of-range values must be quite commonly expected. Two separate features vary with the actual confidence level established. The first is the frequency of occurrence of out-of-range values. The second is the expected maximum and minimum values that exist in the population. To see that this is so, consider a population of 100 numbers ranging uniformly from 0 to 99 without duplication. Take a random sample of 10. Consider two questions: What is the chance of discovering the largest number in the population? and What is the largest value likely to be?

#### Probability of Discovery of Largest Value

Since there are 100 numbers, and only one can be the greatest, on any one random pick there is 1 chance in 100 that the largest number is found. Choosing 10 numbers, each selected at random, from 100 gives 10 chances in 100 for picking the largest number.

By similar reasoning, the chance of finding the largest value in a random sample of, say, 20, is 20%, as shown in Table 7.1.

**TABLE 7.1 Probability of finding largest value for several numbers of picks.**

| Number of picks | Probability in % |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 5 | 5 |
| 10 | 10 |
| 15 | 15 |

## Most Likely High and Low Values

But what is the largest *value* likely to be found? When making the random pick, any values at all could be chosen, each being equally likely. In this example, 10 numbers from 100 are selected (10% of the population), so every number in the population has a 10% chance of being chosen. But what is the most likely value to pick?

Imagine if numbers are selected one at a time at random and a running average of the values picked is kept. Since any number is as likely to be picked as any other, the running average is simply going to approach the average value of all the numbers in the population. If picking continues long enough, all of the numbers are chosen with equal frequency. Added together and divided by the number of picks, the result is the population average value.

In this example, the mean value of the population is 50. Does this mean that 50 is the most likely number to pick? Not exactly. There is only a 1% chance of actually choosing the value 50 in any single pick. If 10% of the population is chosen, the number 50 has a 10% chance of being in the sample. However, what it can be interpreted to mean is that if the choice of one number at random were repeated many times, the numbers chosen would seem to cluster around 50. (There would be as many values of 50 and above as there are below 50, and, on average, they would be as far above as below.) In this sense, 50 indicates the center of the cluster, and so measures the center of the place where the numbers tend to group together. That, indeed, is why the mean is called a “measure of central tendency” in statistics.

What happens when two numbers are picked, paying attention to which is larger and which is smaller? With two numbers selected, it is certain that one is larger than the other (since the population comprises the numbers 1 through 100 without duplicates). By reasoning similar to the single-number pick, the upper value will tend to be halfway between the lower value picked (whatever that is) and the largest number available (100). Similarly, the lower value will tend to be halfway between the higher value picked (whatever that is) and the lowest number available (1). So the two numbers picked will split the range into three parts. Because each value has a tendency to be as far as it can both from its neighbor, and from the extreme values in the range (1 and 100), the separations will be equal in size. In other words, the tendency for two numbers will be to split the range into three equal parts. In this example, for two choices, the expected values are about 33 and 67.

This reasoning can be extended for any number of picks where the order of the picked

values is noted. The expected values are exactly the points that divide the range into $n + 1$ equally sized subranges (where $n$ is the number of picks).

Table 7.2 shows the expected high and low values for a selection of numbers of picks. As the sample size increases, the expected value of the highest value found gets closer and closer to the maximum value of the population. Similarly, with increased sample size, the expected value of the lowest value found in the sample approaches the low value in the population.

**TABLE 7.2 Expected values for various choices.**

| Number of picks | Expected low value | Expected high value |
|---|---|---|
| 1 | 50 | 50 |
| 2 | 33 | 67 |
| 5 | 17 | 83 |
| 10 | 9 | 91 |
| 15 | 6 | 94 |
| 20 | 5 | 95 |

In the example, the population's extreme values are 1 and 100. Table 7.2 shows how the expected high and low values change as the number of picks changes. As the sample size increases, indicated by the number of picks, so the difference between the expected values and the extreme values in the population gets smaller. For instance, the upper-range difference at 10 picks is 100 – 91 = 9, and at 20 picks is 100 – 95 = 5. The lower range difference at 10 picks is 9 – 1 = 8, and at 20 picks is 5 – 1 = 4. (The apparent difference in the upper and lower range is due to rounding off the values. The upper and lower expected values are actually symmetrically located in the range.)

## Out-of-Range Values and the PIE

The examples just given are considerably simplified. For real-world variables with real-world distributions, things are far more complex. Actual probabilities and expected

values depend very much on the true distribution of a variable, among other things. This is, in any case, complicated by the fact that distributions may change over time. This example is true for a rectangular distribution (one in which every value that can occur does so with equal probability) and where no values are duplicated. In the example, the size of the population was also known, which makes determining probabilities easy.

While the probabilities vary considerably with distribution, population size, and other factors, the principles do not:

Some maximum and minimum values will be detected in a sample.

- The discovered maximum and minimum define the range of the sample.
- In the population there is always some chance of encountering an out-of-range value.
- Some specific confidence that the sample is representative of the population can be determined.
- The smaller the chance that the sample is representative, the larger the chance of encountering an out-of-range value, and the larger the gap is likely to be between sample range limit and the population limit.

That is, the less representative the sample, the more chance there is of encountering an out-of-range value in the population. And when an out-of-range value is found, the less representative the sample, the greater the expected difference will be between the sample maximum or minimum and the out-of-range value.

Knowing the confidence level that the data sample is representative does give some indication of how likely an out-of-range value is to be found in the population, and how large the gap between the detected limits and the out-of-range value might be. Having these estimates enables the normalization process to be adjusted to take account of the expected frequency of out-of-range values and the expected true range of the population values. For instance, if the sample confidence is relatively low, then many out-of-range values covering a large range can be expected. If sample confidence is high, few out-of-range values will be expected, and those that are will cover a narrower range.

To summarize: The representative sample selected to create the PIE has, for every variable, high and low values. When building the PIE, no values exceeding this range will be found, since it is the sample that produced the maximum and minimum values. The confidence level that the sample is representative gives an indication of the probability of meeting out-of-range values. The confidence level also indicates the probable size of the gap between discovered maximum or minimum, and any out-of-range value. With the frequency of occurrence and the gap size estimated, the normalization process in the PIE can be constructed accordingly.

### 7.1.4 Out-of-Range Values When Training

During training, the PIE is already built and in place. The PIE-I takes raw data values from the training sample and translates them into a prepared state for use by the modeling tool. What happens when the PIE-I finds an out-of-range value? As yet there has been no discussion of a method of dealing with out-of-range values. What would happen if out-of range values are not normalized into a 0–1 range, but passed through outside the normalized range? That is, an input value larger than the sample maximum would translate into a value larger than 1, perhaps 1.2. Similarly, input values smaller than the sample minimum will translate into values less than 0, maybe –0.2. (The purpose of the discussion, of course, is to examine the problems that could occur to discover how best to avoid them.)

#### Consequences of Ignorance I

One “solution” to an out-of-range value (adopted by some commercial modeling tools) is to simply ignore the whole instance (record) if any one of the values is out-of-range. This also takes care of the missing-value problem. (Missing values are treated as out-of-range too.) This effectively reduces the size of the sample by ignoring any data points that do not fit within the specified parameters. There are two notable problems with this approach.

The first, and less significant, problem is that reducing the number of instances in the sample reduces the level of confidence that the sample represents the population. Discarding instances is literally discarding information! Discarding, or ignoring, instances effectively reduces the size of the training set. A model created using the reduced training set cannot be as effective as one built with a more representative data set. If this were the only problem, it is easily remedied by adding more data to the training set if it is available. Adding more data again increases confidence that the sample is representative. On the other hand, if more data is not available, it may be that the information in the discarded instances is much needed, and discarding them is damaging to training.

A second, and potentially more serious problem, is introducing bias. Unless the out-of-range values occur in a truly random pattern, then obviously they do not occur at random. If they do not occur at random, then they must occur with some sort of pattern. Deleting or ignoring out-of-range instances then necessarily removes them according to some pattern. Removing instances in a pattern prevents the modeling tool from seeing the pattern. This removal of a pattern from the sample introduces distortion, or bias, to the sample. The bias can be anything from slightly damaging to disastrous—with no way to determine which! This problem is so potentially severe and undetectable that attempts must be made to avoid it at all costs.

Imagine (as really happened) using such a tool for building a model of mortgage applicants. The training sample had applicants with salaries up to, say, $100,000. When

the model was run, this method ignored all applicants with salaries greater than $100,000. But the null score was interpreted as no score, and the mortgage company interpreted no score as a bad score! Until discovered (which didn't take long), this method of dealing with out-of-range variables was (to say the least) problematic. In practice, of course, it rendered the model virtually useless.

## Consequences of Ignorance II

Another approach ignores the fact that the normalized range has been exceeded. It says, "Let normalized values fall outside the range if necessary." The assumptions about state space being *unit* state space will no longer hold, but this is not always a major concern since state space may only be a conceptual device for many modeling methods. Most modeling tools have at least some capacity to handle numbers outside the normalized range. But how do they handle them? And does it make a difference to the quality of the model?

Some methods do use a unit state space model. Where this is the case, these will have to deal with the out-of-limit values in a way that keeps them inside unit state space. One method is to "clip" the values that fall outside the range. If greater than 1, assign 1. If less than 0, assign 0. The problem with this method is the underlying assumption that numbers that fall outside the range are in some way equivalent to numbers that fall at the limit of the range. This ignores the fact that the numbers falling outside the range *are* in some way different and carry information to that effect. This vital information is thrown away.

Worse than throwing information away is what happens to the limit values if there is a difference that the model should reflect between limit values and out-of-range values. The limit value's information content is distorted because the model will not be able to distinguish between range limit and out-of-range values. The range limit value meaning will have to be distorted to reflect whatever aggregate meaning the out-of-range values carry, too. Projecting the information content from several values onto a single value distorts the information content of the limit value.

For example, if the out-of-range values extend up to 1.2, the range top value of 1 has to carry an "average" meaning of all the values from 1 to 1.2. Any difference that the model should reflect when the value is, say, 1.1 is lost, merged, as it were, with the meaning carried by the range top value of 1. But worse, if the model is predictive, for instance, when the input value is actually 1, the model will have to predict the "average" response of values 1 through 1.2.

Once again, the problem of bias shows up. If the occurrence of out-of-range values is not in fact random, using exactly the same argument as in the previous section, undetectable bias is introduced into the model. Just as before, the problems this introduces can range from innocuous to disastrous. Bias can invalidate the best model.

In some models, for instance, fraudulent activity falls into this out-of-range category. It is the fraudulent activity that may fall out of the modeled range, since new patterns of fraud constantly evolve. If the fraudulent activity moves some variable instance values out of their limits, and the model is constrained to ignore it, or to “merge” it with other activity, this new activity is indicated as equivalent to whatever the model found to be the activity at the range limit. This may easily be an unjustified assumption.

In one case, this “merging” behavior persuaded one model of insurance claims to assume that all building fires occurring after 9:30 at night, and started in rear rooms, scored pretty well as likely arson! In fact, this model made a number of other erratic inferences, all due to the nature of the insurance claim data set modeled and the tool used.

### 7.1.5 Out-of-Range Values When Testing

When testing models, many of the same problems occur as when training. Testing attempts to discover the applicability of, and limits to, the model. Whether or not the training phase experienced out-of-limit values, if no correction or allowance is made for them, their presence during testing will be dealt with in a similar, cavalier way. In one way or another they will be either ignored or clipped. For all of the reasons discussed above, this will produce a less accurate model output. (The actual output will be numeric, although the final result might be inferences, predictions, or come in some other form, depending on the type of model.)

Testing the model in ways that underestimate its limits and utility is not necessarily damaging, but will lead at least to having less confidence in the model than is perhaps justified. It will certainly help in making, to some extent, erroneous conclusions about the range, utility, and applicability of the model.

However, the model might also appear to be better and more robust than is actually the case. Ignoring instances of data in the test data set because they have out-of-range values, for instance, clearly means that the model is not tested on them. But these are the precise areas in which the model might perform most poorly, and its performance in these areas has to be included in any valid overall performance summary.

### 7.1.6 Out-of-Range Values When Executing

Execution is the time when a predictive model is predicting, an inferential model is inferring, a self-adaptive model is adapting, and so on. Whatever else went before, this is the time when out-of-range values are most likely to appear if they ever will! This is the phase of the data exploration project when the model is likely to be exposed to copious quantities of data, and so has the highest expectation that the fullest range of the data will appear. (It is also the time when real, applicable, and useful results are expected.) For simplicity of discussion, a predictive model will be assumed. The same principles hold for any type of model—predictive, inferential, adaptive, and so on.

A model created by training on data biased by removing problematic instances from the training data will almost certainly still be required to produce predictions for similar problematic instances in the execution data. If predicting fraud, for instance, all instances must be examined. If predicting customer segments, all customers must be predicted. The model is not considered adequate if no predictions are made for instances with problematic data. (Even people earning more than $100,000 may be good mortgage risks!) But if out-of-range values were excluded during training, the model was not exposed to such data during training. There is no reference for making a valid prediction from such data during the execution phase. In any case, the model will be more or less biased, having been trained on biased data. The execution data that the model is required to perform on will not be biased. Whatever bias is included in the model will result in biased predictions.

If, on the other hand, the out-of-range values were “trimmed” off to the limiting values during training, when the model does experience such values, they will have to be trimmed again, leading to poor predictions for any limiting conditions.

Possibly the worst scenario is that untrimmed variable values are allowed into the model. When this happens, the model is driven outside of the range of data on which it trained. In this case the model will, of course, produce predictions, but predictions that are based on no evidence. When the model is driven into areas that are outside the boundaries of the state space on which it trained, almost no valid predictions can be made. We can speculate, for instance, about the weight of 20-foot-tall human beings. Whatever extrapolation we might make, the truth is that there is no evidence to base a prediction on, for such a creature probably could not exist. Whether or not such a being could exist, and what its weight might be, is pure speculation. So it is too when a model is driven beyond the limits on which it trained.

Clipping values leaves the model no way to detect if the instance values are changing—at least at the limits of behavior. It is often the case that the distribution of the data is not stationary. Nonstationarity of a distribution simply means that the distribution does not remain constant, usually over time. The user of the model needs to monitor this, among many other things, during run time anyway. This is not a part of data preparation, but the preparation technique should at least provide support to make the monitoring easier.

### 7.1.7 Scaling Transformations

The discussion so far has looked at the issues surrounding finding the maximum and minimum values for each variable in a sample. Clearly, knowing the maximum and minimum values somehow allows the actual value to be scaled, or normalized, into the range 0–1. A way of doing this is to use a transforming expression that takes the input value, and, knowing the maximum and minimum values, squashes the input value into the required output range. An easy way to actually do this is with the *linear scaling transform*.

The actual expression is very straightforward:

$$v_n = \frac{v_i - \min(v_1 \ldots v_n)}{\max(v_1 \ldots v_n) - \min(v_1 \ldots v_n)}$$

where

$v_n$ is normalized value

$v_i$ is instance value

This expression takes any value and transforms it into another number. If the input value is inside the limits, the output will be between 0 and 1. Any value outside the limits will fall outside the 0–1 range, presenting a modeler with all of the problems just discussed.

## Using Linear Scaling for Normalization

Although many of the problems of dealing inadequately with out-of-range values have been discussed, and the simplest method of normalizing values has been found wanting, it is still the place to start. Linear scaling is a simple, straightforward technique to use for normalizing the range of numeric values. Its big advantage is that it introduces no distortion to the variable distribution. It involves only discovering the maximum and minimum values for the range of the variable, and then finding where within the range a particular instance value falls. The formula for achieving this is given above. Given this formula, any instance value can be plugged in, and a normalized value computed. There is a one-to-one relationship between the original instance value and the normalized value. Given two instance values, with the first being twice the second, when they are normalized, the first normalized value will still be twice the second. This is true wherever in the range of the variable the two instance values occur.

The relationship between the instance values and the normalized values is called linear because if the two sets of values are plotted on a graph, the result is a straight line—as shown in Figure 7.1.

| Number | Normalized |
|---|---|
| 40.12 | 0.56 |
| 31.29 | 0.42 |
| 43.85 | 0.63 |
| 40.60 | 0.57 |
| 54.21 | 0.80 |
| 46.94 | 0.68 |
| 28.06 | 0.36 |
| 60.97 | 0.91 |
| 57.39 | 0.85 |
| 66.09 | 1.00 |
| 6.45 | 0.00 |
| 9.53 | 0.05 |
| 52.83 | 0.78 |
| 47.44 | 0.69 |
| 21.00 | 0.24 |
| 43.31 | 0.62 |
| 14.40 | 0.13 |
| 37.42 | 0.52 |
| 16.47 | 0.17 |
| 61.51 | 0.92 |

| | |
|---|---|
| Maximum | 66.09 |
| Minimum | 6.45 |
| Difference | 59.64 |

$$v_{norm} = \frac{v_j - \min(v_1 \ldots v_n)}{\max(v_1 \ldots v_n) - \min(v_1 \ldots v_n)}$$

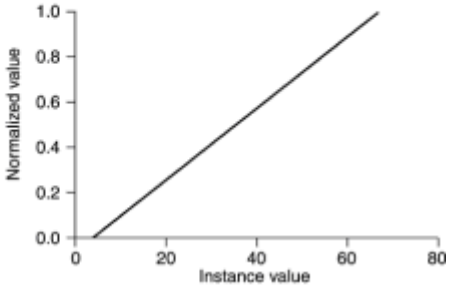


**Figure 7.1** Linear scaling produces a linear relationship between instance values and normalized values.

Linear scaling works well when the maximum and minimum values are known. During data preparation, this presents a problem. The maximum and minimum values of the *sample* are known, but the true population maximum and minimum may be unknowable. As just discussed, when using the model with real-world data, it is very likely that instance values outside the sample range will be encountered. Linear scaling normalization will translate these values into numbers that fall outside the 0–1 range.

In spite of its shortcomings, which only occur at the limit of the range, linear scaling has a great strength in that it introduces no distortion in the translated values. Whatever information is in the original values is preserved unmodified in the normalized values. This is an important feature that needs to be preserved, if at all possible. If the problems that occur at the limits can be dealt with, linear scaling works well.

## Making Room for Out-of-Range Values

In order to deal with the out-of-range problems, the PIE needs a method of dealing with the limit problems of linear scaling. The linear scaling transformation gives linear normalization over all of its range, but must be modified to somehow include out-of-range values. One way to do this is to reduce the part of the transformed range that holds in-range values. There is then room to squeeze the out-of-range values into space left at the upper and lower ends, still leaving some differentiation between them. But how can this be done? Theoretically, there is some chance, however small, that an arbitrarily far out-of-range number will be encountered at either end of the range. How can what is potentially an infinite range of numbers be squashed into a finite part of a 0–1 range, especially with most of the 0–1 range given over to linear scaling? Fortunately, exactly such a transform does exist, and it forms the basis of *softmax scaling*. This key transform is called the *logistic function*. Both softmax scaling and the logistic function are examined shortly. But first, what exactly is it that needs to be done?

The optimal form of normalizing transformation, if it could be guaranteed never to go out of range, is linear. Of course, that's just the problem—it can be guaranteed to go out of range with some degree of confidence. However, if we can measure, or make assumptions about, the distribution of the variable, we can then make inferences about the distance between the sample limit and the population limit. Doing this allows choosing some appropriate part of the range to be linear—and some appropriate part to accommodate the out-of-range values.

Since the idea is that the translation is linear over some part of the range, the question is, how much of the range should be linear? The sample being used for building the PIE is selected with some degree of confidence. It is this confidence that can be used to determine what part of the 0–1 range is to be held linear. The size of the expected out-of-range gap is directly proportional to the degree of confidence that there will be out-of-range values. If, for example, the selected confidence level was 98%, then 98% of the range 0–1 will be linear. The selected linear part of the range is centered, so that the linear translation range becomes, for a 98% confidence level, 0.01–0.99. The linear part of the range is squashed by 2%. The balance from the 98%, or 2%, is evenly spaced at the top and bottom of the range. This leaves 1% at each end of the range for squeezing in the out-of-limit numbers. Figure 7.2 illustrates squashing the linear part of the range.

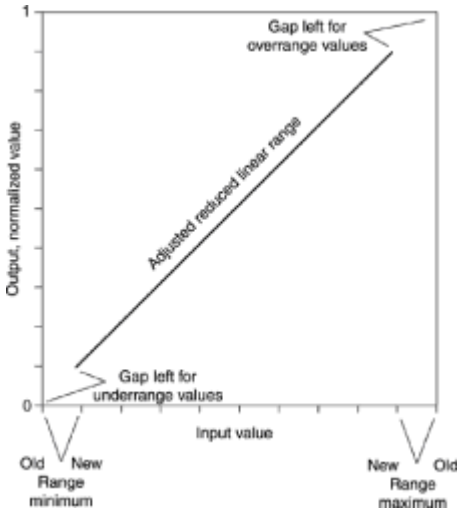


**Figure 7.2** The linear part of the range is "compressed" so that it covers a smaller part of the output range. The "gaps" left at the top and bottom allow space in which to compress the out-of-range values.

## Squashing the Out-of-Range Numbers

The problem that now remains is to fit the out-of-range numbers, potentially extending to infinity, into the minute space left for them. Consider the upper limit. First, it is important to

realize that for numbers larger than the limit, the greater a number, the less likely it is that any such value will be found. When the sample was originally taken, some degree of confidence was established that the largest value had been found. Larger numbers are possible, even likely, but, as previously discussed, the greater the difference between the limit value and any larger value, the less likely it is that it will be encountered.

The transformation is made such that as the difference between the limit and the out-of-limit value grows, the smaller the increase toward the end of the range. Larger numbers produce proportionally smaller differences, and an infinitely large number produces ultimately infinitesimally small differences. In Chapter 6 it was noted that by increasing precision, it is always possible to indicate more locations on the number line. This allows an infinite number of out-of-range numbers to be mapped into space left for them. If such a transform is developed, it can be used to squash the out-of-range values above and below the linear part into the space left for them.

Looking at the upper range, a mathematical function is needed such that as the difference between the limit and overlimit values gets larger, the value increases toward, but never reaches, some boundary. Whatever its limits, the output of the squashing function can itself then be linearly squashed to fit into the gap left for it. In looking for such a function, a reciprocal makes a good starting point. A reciprocal of a number is simply one divided by the number. It starts with a value of one, and as the input number gets larger, the output value gets smaller and smaller, reaching toward, but never getting to, 0. To have this transform move in the opposite direction, subtract it from 1. It becomes $1 - 1/v$. Table 7.3 shows the output values for various inputs.

**TABLE 7.3 Values of upper-range squashing function.**

| v | 1/*v* | 1 – 1/*v* |
|---|---|---|
| 1 | 1.000 | 0.000 |
| 2 | 0.500 | 0.500 |
| 3 | 0.333 | 0.667 |
| 5 | 0.200 | 0.800 |
| 8 | 0.125 | 0.875 |
| 13 | 0.077 | 0.923 |

| | | |
|---|---|---|
| 21 | 0.048 | 0.952 |
| 34 | 0.029 | 0.971 |
| 55 | 0.018 | 0.982 |
| 89 | 0.011 | 0.989 |

So 1 – 1/*v* starts at 0 and moves toward 1, never quite reaching it, regardless of how large a number is input. This is shown graphically in Figure 7.3(a).

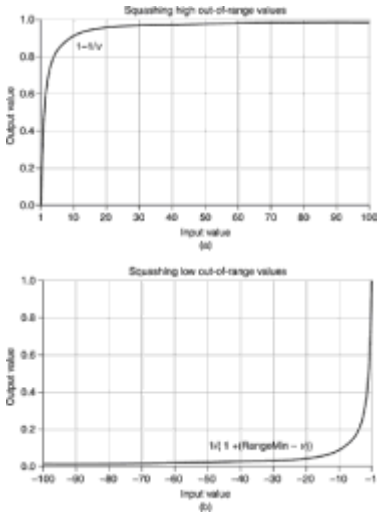


**Figure 7.3** Values of 1 – 1/*v* for the range of inputs 1 through 100 (a) and values of 1/ (1+(RangeMin – *v*)) for the range of inputs –100 through –1(b).

Squashing out-of-range values into the lower space left can use the same transform. This time the out-of-range difference is discovered (RangeMin – *v*) and 1 is added to it to ensure that the difference can never be less than 1. This time *v* is the lower-range difference, but Table 7.3, in column 1/*v*, shows the values for this starting at 1 and decreasing toward, but never reaching, 0. Figure 7.3(b) shows this graphically. These two curves, one for the upper out-of-range values and one for the lower out-of-range values, need to be squashed and attached to the linear part of the transform.

Taking the linear part of the range and adding the upper and lower transforms for the out-of-range values produces a curve. The result will be a sort of “S” curve that is linear over most of the range, but squashes the over- and undervalues into the remaining space. Figure 7.4 shows the same curves squashed into the range. (The amount of the scale

allocated for squashing out-of-range values is highly exaggerated to illustrate the point.)

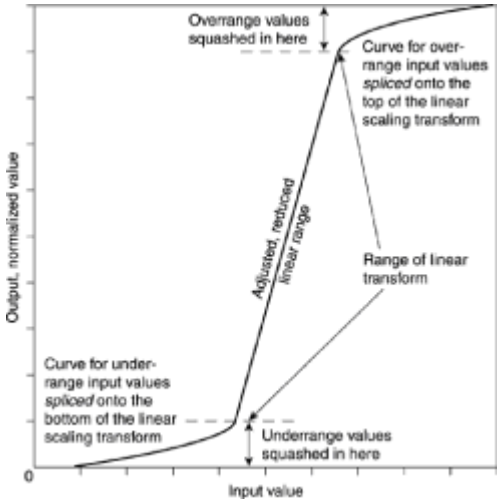


**Figure 7.4** The transforms for squashing overrange and underrange values are attached to the linear part of the transform. This composite "S"-shaped transform translates most of the values linearly, but also transforms any out-of-range values so that they stay within the 0–1 limits of the range.

This sort of "S" curve can be constructed to serve the purpose. Writing computer code to achieve this is somewhat cumbersome. The description shows very well the sort of effect that is needed, but fortunately there is a much easier and more flexible way to get there.

## 7.1.8 Softmax Scaling

Softmax scaling is so called because, among other things, it reaches "softly" toward its maximum value, never quite getting there. It also has a linear transform part of the range. The extent of the linear part of the range is variable by setting one parameter. It also reaches "softly" toward its minimum value. The whole output range covered is 0–1. These features make it ideal as a transforming function that puts all of the pieces together that have been discussed so far.

### The Logistic Function

It starts with the *logistic function*. The logistic function can be modified to perform all of the work just described, and when so modified, it does it all at once so that by plugging in a variable's instance value, out comes the required, transformed value.

An explanation of the workings of the logistic function is in the Supplemental Material section at the end of this chapter. Its inner workings are a little complex, and so long as what needs to be done is clear (getting to the squashing "S" curve), understanding the logistic function itself is not necessary. The Supplemental Material can safely be skipped.

The explanation is included for interest since the same function is an integral part of neural networks, mentioned in Chapter 10. The Supplemental Material section then explains the modifications necessary to modify it to become the softmax function.

### 7.1.9 Normalizing Ranges

What does softmax scaling accomplish in addressing the problems of range normalization? The features of softmax scaling are as follows:

The normalized range is 0–1. It is the nature of softmax scaling that no values outside this range are possible. This keeps all normalized values inside unit state space boundaries. Since the range of input values is essentially unlimited and the output range is limited, unit state space, when softmax is normalized, is essentially infinite.

- The extent of the linear part of the normalized range is directly proportional to the level of confidence that the data sample is representative. This means that the more confidence there is that the sample is representative, the more linear the normalization of values will be.

- The extent of the area assigned for out-of-range values is directly proportional to the level of uncertainty that the full range has been captured. The less certainty, the more space to put the expected out-of-range values when encountered.

- There is always some difference in normalized value between any two nonidentical instance values, even for very large extremes.

As already discussed, these features meet many needs of a modeling tool. A static model may still be presented with out-of-range values where its accuracy and reliability are problematic. This needs to be monitored separately during execution time. (After all, softmax squashing them does not mean that the model knows what to do with them—they still represent areas of state space that the model never visited during training.) Dynamic models that continuously learn from the data stream—such as continuously learning, self-adaptive, or response-adaptive models—will have no trouble adapting themselves to the newly experienced values. (Dynamic models need to interact with a dynamic PIE if the range or distribution is not stationary—not a problem to construct if the underlying principles are understood, but not covered in detail here.)

At the limits of the linear normalization range, no modeling tool is required to aggregate the effect of multiple values by collapsing them into a single value (“clipping”).

Softmax scaling does the least harm to the information content of the data set. Yet it still leaves some information exposed for the mining tools to use when values outside those within the sample data set are encountered.

## 7.2 Redistributing Variable Values

Through normalization, the range of values of a variable can be made to always fall between the limits 0–1. Since this is a most convenient range to work with, it is assumed from here on that all of a variable's values fall into this range. It is also assumed that the variables fall into the linear part of the normalized range, which will be true during data preparation.

Although the range is normalized, the distribution of the values—that is, the pattern that exists in the way discrete instance values group together—has not been altered. (Distributions were discussed in Chapters 2 and 5.) Now attention needs to be turned to looking at the problems and difficulties that distributions can make for modeling tools, and ways to alleviate them.

### 7.2.1 The Nature of Distributions

Distributions of a variable only consist of the values that actually occur in a sample of many instances of the variable. For any variable that is limited in range, the count of possible values that can exist is in practice limited.

Consider, for example, the level of indebtedness on credit cards offered by a particular bank. For every bank there is some highest credit line that has ever been offered to any credit card customer. Large perhaps, but finite. Suppose that maximum credit line is $1,000,000. No credit card offered by this bank can possibly have a debit balance of more than $1,000,000, nor less than $0 (ignoring credit balances due, say, to overpayment). How many discrete balance amounts are possible? Since the balance is always stated to the nearest penny, and there are 100 pennies in a dollar, the range extends from 0 pennies to 100,000,000 pennies. There are no more than 100,000,000 possible discrete values in the entire range.

In general, for any possible variable, there is always a particular resolution limit. Usually it is bounded by the limits of accuracy of measurement, use, or convention. If not bounded by those, then eventually the limits of precision of representation impose a practical limit to the possible number of discrete values. The number may be large, but it is limited. This is true even for softmax normalization. If values sufficiently out of range are passed into the function, the truncation that any computer requires eventually assigns two different input values to the same normalized value. (This practical limitation should not often occur, as the way in which the scale was constructed should preclude many far out-of-range values.)

However many value states there are, the way the discrete values group together forms patterns in the distribution. Discrete value states can be close together or far apart in the range. Many variables permit identical values to occur—for example, for credit card balances, it is perfectly permissible for multiple cards to have identical balances.

A variable's values can be thought of as being represented in a one-dimensional state space. All of the features of state space exist, particularly including clustering of values. In some parts of the space the density will be higher than in other parts. Overall there will be some mean density.

### 7.2.2 Distributive Difficulties

One of the problems of distribution is outlying values or outlying clumps. (Figure 2.5 illustrates this.) Some modeling techniques are sensitive only to the linear displacement of the value across the range. This only means that the sensitivity remains constant across the range so that any one value is as "important" as any other value. It seems reasonable that 0.45 should be as significant as 0.12. The inferences to be made may be different—that is, each discrete value probably implies a different predicted value—but the fact that 0.45 has occurred is given the same weight as the fact that 0.12 has occurred.

Reasonable as this seems, it is not necessarily so. Since the values cluster together, some values are more common than others. Some values simply turn up more often than others. In the areas where the density is higher, values occurring in that area are more frequent than those values occurring in areas of lower density. In a sense, that is what density is measuring—frequency of occurrence. However, since some values are more common than others, the fact that an uncommon one has occurred carries a "message" that is different than a more common value. In other words, the weighting by frequency of specific values carries information.

To a greater or lesser degree, density variation is present for almost all variables. In some cases it is extreme. A binary value, for instance, has two spikes of extremely high density (one for the "0" value and one for the "1" value). Between the spikes of density is empty space. Again, most alpha variables will translate into a "spiky" sort of density, each spike corresponding to a specific label.

Figure 7.5 illustrates several possible distributions. In Figure 7.5(d) the outlier problem is illustrated. Here the bulk of the distribution has been displaced so that it occupies only half of the range. Almost half of the range (and half of the distribution) is empty.

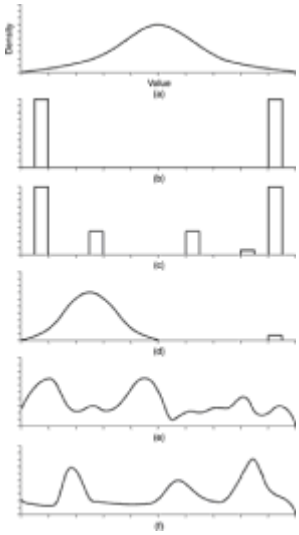

**Figure 7.5** Different types of distributions and problems with the distribution of a variable's values across a normalized range: normal (a), bimodal or binary variable (b), alpha label (c), normal with outlier (d), typical actual variable A (e), and typical actual variable B (f). All graphs plot value (x) and density (y).

Many, if not most, modeling tools, including some standard statistical methods, either ignore or have difficulty with varying density in a distribution. Many such tools have been built with the assumption that the distribution is normal, or at least regular. When density is neither normal nor regular, as is almost invariably the case with real-world data sets—particularly behavioral data sets—these tools cannot perform as designed. In many cases they simply are not able to "see" the information carried by the varying density in the distribution. If possible, this information should be made accessible.

When the density variation is dissimilar between variables, the problem is only intensified. Between-variable dissimilarity means that not only are the distributions of each variable irregular, but that the irregularities are not shared by the two variables. The distributions in Figure 7.5(e) and 7.5(f) show two variables with dissimilar, irregular distributions.

There are tools that can cope well with irregular distributions, but even these are aided if the distributions are somehow regularized. For instance, one such tool for a particular data set could, when fine-tuned and adjusted, do just as well with unprepared data as with prepared data. The difference was that it took over three days of fine-tuning and adjusting by a highly experienced modeler to get that result—a result that was immediately available with prepared data. Instead of having to extract the gross nonlinearities, such tools can then focus on the fine structure immediately. The object of data preparation is to expose the maximum information for mining tools to build, or extract, models. What can be done to adjust distributions to help?

### 7.2.3 Adjusting Distributions

The easiest way to adjust distribution density is simply to displace the high-density points into the low-density areas until all points are at the mean density for the variable. Such a process ends up with a rectangular distribution. This simple approach can only be completely successful in its redistribution if none of the instance values is duplicated. Alpha labels, for instance, all have identical numerical values for a single label. There is no way to spread out the values of a single label. Binary values also are not redistributed using this method. However, since no other method redistributes such values either, it is this straightforward process that is most effective.

In effect, every point is displaced in a particular direction and distance. Any point in the variable's range could be used as a reference. The zero point is as convenient as any other. Using this as a reference, every other point can be specified as being moved away from, or toward, the reference zero point. The required displacements for any variable can be graphed using, say, positive numbers to indicate moving a point toward the "1," or increasing their value. Negative numbers indicate movement toward the "0" point, decreasing their value.

Figure 7.6 shows a distribution histogram for the variable "Beacon" included on the CD-ROM in the CREDIT data set. The values of Beacon have been normalized but not redistributed. Each vertical bar represents a count of the number of values falling in a subrange of 10% of the whole range. Most of the distribution shown is fairly rectangular. That is to say, most of the bars are an even height. The right side of the histogram, above a value of about 0.8, is less populated than the remaining part of the distribution as shown by the lower height bars. Because the width of the bars aggregates all of the values over 10% of the range, much of the fine structure is lost in a histogram, although for this example it is not needed.

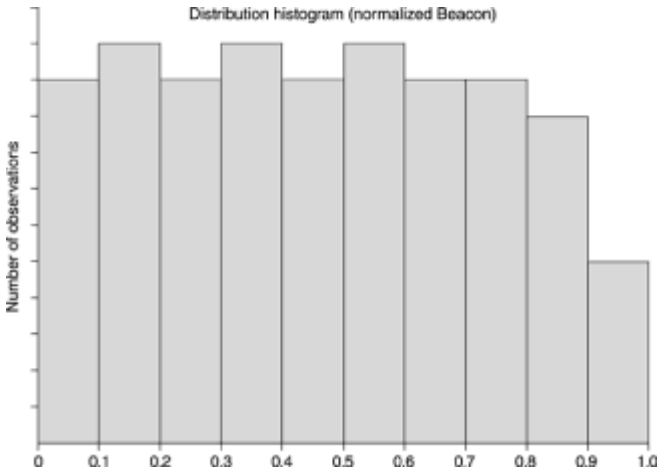


**Figure 7.6** Distribution histogram for the variable Beacon. Each bar represents 10% of the whole distribution showing the relative number of observations (instances) in each bar.

Figure 7.7 shows a displacement graph for the variable Beacon. The figure shows the movement required for every point in the distribution to make the distribution more even. Almost every point is displaced toward the “1” end of the variable’s distribution. Almost all of the displaced distances being “+” indicates the movement of values in that direction. This is because the bulk of the distribution is concentrated toward the “0” end, and to create evenly distributed data points, it is the “1” end that needs to be filled.

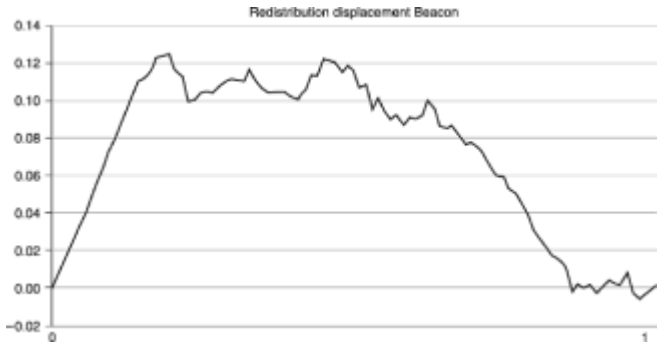


**Figure 7.7** Displacement graph for redistributing the variable Beacon. The large positive “hump” shows that most of the values are displaced toward the “1” end of the normalized range.

Figure 7.8 shows the redistributed variable’s distribution. This figure shows an almost perfect rectangular distribution.

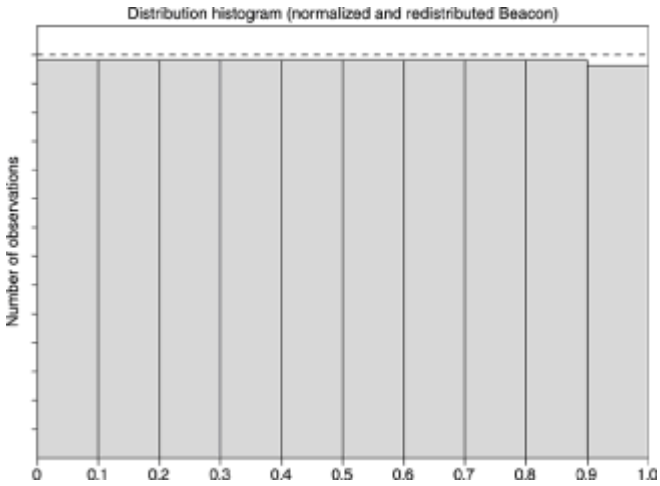


**Figure 7.8** The distribution of Beacon after redistribution is almost perfectly rectangular. Redistribution of values has given almost all portions of the range an equal number of instances.

Figure 7.9 shows a completely different picture. This is for the variable DAS from the same data set. In this case the distribution must have low central density. The points low in the range are moved higher, and the points high in the range are moved lower. The positive curve on the left of the graph and the negative curve to the right show this clearly.

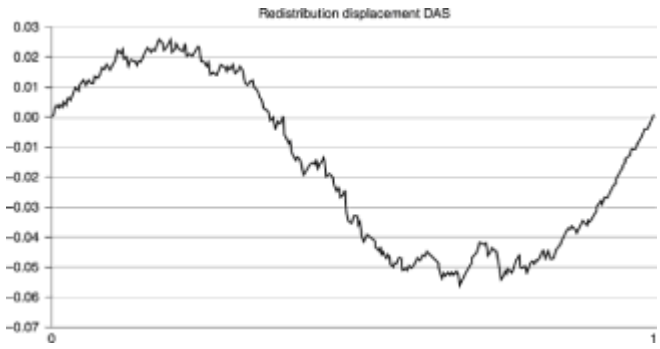


**Figure 7.9** For the variable DAS, the distribution appears empty around the middle values. The shape of the displacement curve suggests that some generating phenomenon might be at work.

A glance at the graph for DAS seems to show an artificial pattern, perhaps a modified sine wave with a little noise. Is this significant? Is there some generating phenomenon in the real world to account for this? If there is, is it important? How? Is this a new discovery? Finding the answers to these, and other questions about the distribution, is properly a part of the data survey. However, it is during the data preparation process that they are first "discovered."

### 7.2.4 Modified Distributions

When the distributions are adjusted, what changes? The data set CARS (included on the accompanying CD-ROM) is small, containing few variables and only 392 instances. Of the variables, seven are numeric and three are alpha. This data set will be used to look at what the redistribution achieves using "before" and "after" snapshots. Only the numeric variables are shown in the snapshots as the alphas do not have a numeric form until after numeration.

Figures 7.10(a) and 7.10(b) show box and whisker plots, the meaning of which is fairly self-explanatory. The figure shows maximum, minimum, median, and quartile information. (The median value is the value falling in the middle of the sequence after ordering the values.)

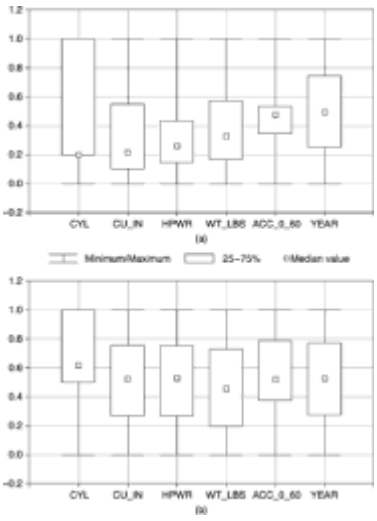


**Figure 7.10** These two box and whisker plots show the before and after redistribution positions—normalized only (a) and normalized and redistributed (b)—for maximum, minimum, and median values.

Comparing the variables, before and after, it is immediately noticeable that all the median values are much more centrally located. The quartile ranges (the 25% and 75% points) have been far more appropriately located by the transformation and mainly fall near the 25% and 75% points in the range. The quartile range of the variable “CYL” (number of cylinders) remains anchored at “1” despite the transformation—why? Because there are only three values in this field—“4,” “6,” and “8”—which makes moving the quartile range impossible, as there are only the three discrete values. The quartile range boundary has to be one of these values. Nonetheless, the transformation still moves the lower bound of the quartile range, and the median, to values that better balance the distribution.

Figures 7.11(a) and 7.11(b) show similar figures for standard deviation, standard error, and mean. These measures are normally associated with the Gaussian or normal distributions. The redistributed variables are not translated to be closer to such a distribution. The translation is, rather, for a rectangular distribution. The measures shown in this figure are useful indications of the regularity of the adjusted distribution, and are here used entirely in that way. Once again the distributions of most of the variables show considerable improvement. The distribution of “CYL” is improved, as measured by standard deviation, although with only three discrete values, full correction cannot be achieved.

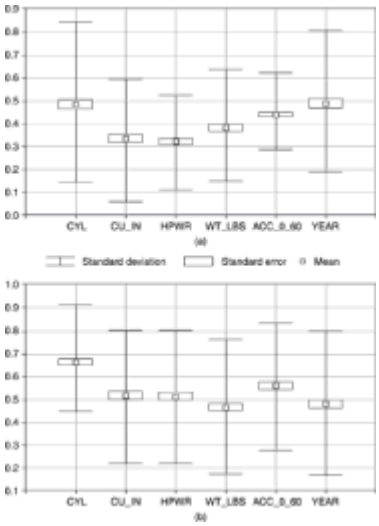


**Figure 7.11** These two box and whisker plots show the before and after redistribution positions—normalized only (a) and normalized and redistributed (b)—for standard deviation, standard error, and mean values.

Table 7.4 shows a variety of measures about the variable distributions before and after transformation. “Skewness” measures how unbalanced the distribution is about its center point. In every case the measure of skewness is less (closer to 0) after adjustment than before. In a rectangular distribution, the quartile range should cover exactly half the range (0.5000) since it includes the quarter of the range immediately above and below the median point. In every case except “Year,” which was perfect in this respect to start with, the quartile range shows improvement.

**TABLE 7.4 Statistical measures before and after adjustment.**

| BEFORE: | Mean | Median | Lower quartile | Upper quartile | Quartile range | Std. dev. | Skew-ness |
|---|---|---|---|---|---|---|---|
| CYL | 0.4944 | 0.2000 | 0.2000 | 1.0000 | 0.8000 | 0.3412 | 0.5081 |
| CU_IN | 0.3266 | 0.2145 | 0.0956 | 0.5594 | 0.4638 | 0.2704 | 0.7017 |
| HPWR | 0.3178 | 0.2582 | 0.1576 | 0.4402 | 0.2826 | 0.2092 | 1.0873 |
| WT_LBS | 0.3869 | 0.3375 | 0.1734 | 0.5680 | 0.3947 | 0.2408 | 0.5196 |