2
Types of Data

Now that we have a basic introduction to the world of data science and understand why the field is so important, let's take a look at the various ways in which data can be formed. Specifically, in this chapter we will look at the following topics:

- Structured versus unstructured data
- Quantitative versus qualitative data
- The four levels of data

We will dive further into each of these topics by showing examples of how data scientists look at and work with data. This chapter is aimed to familiarize ourselves with the fundamental ideas underlying data science.

Flavors of data

In the field, it is important to understand the different flavors of data for several reasons. Not only will the type of data dictate the methods used to analyze and extract results, knowing whether the data is unstructured or perhaps quantitative can also tell you a lot about the real-world phenomenon being measured.

We will look at the three basic classifications of data:

- Structured vs unstructured (sometimes called organized vs unorganized)
- Quantitative vs qualitative
- The four levels of data

The first thing to pay attention to is my use of the word *data*. In the last chapter, I defined data as merely being a collection of information. This vague definition exists because we may separate data into different categories and need our definition to be loose.

The next thing to remember while we go through this chapter is that for the most part, when I talk about what type of *data* this is, I will refer to either a *specific characteristic* of a dataset or to the *entire dataset* as a whole. I will be very clear about which one I refer to at any given time.

Why look at these distinctions?

It might seem worthless to stop and think about what type of data we have before getting into the fun stuff, like statistics and machine learning, but this is arguably one of the most important steps you need to take to perform data science.

Consider an example where we are looking at election results for a county. In the dataset of people, there is a "race" column that is denoted via an identifying number to save space. For example perhaps caucasian is denoted by 7 while Asian American is 2. Without understanding that these numbers are not actually ordered numbers like we think about them (where 7 is greater than 2 and therefore caucasian is "greater than" Asian American) we will make terrible mistakes in our analysis. Discuss

The same principle applies to data science. When given a dataset, it is tempting to jump right into exploring, applying statistical models, and researching the applications of machine learning in order to get results faster. However, if you don't understand the type of data that you are working with, then you might waste a lot of time applying models that are known to be ineffective with that specific type of data.

When given a new dataset, I always recommend taking about an hour (usually less) to make the distinctions mentioned in the following sections.

Structured versus unstructured data

The distinction between structured and unstructured data is usually the first question you want to ask yourself about the *entire* dataset. The answer to this question can mean the difference between needing three days or three weeks of time to perform a proper analysis.

The basic breakdown is as follows (this is a rehashed definition of organized and unorganized data in the first chapter):

- **Structured (organized) data**: This is data that can be thought of as observations and characteristics. It is usually organized using a table method (rows and columns).
- **Unstructured (unorganized) data**: This data exists as a free entity and does not follow any standard organization hierarchy.

Here are a few examples that could help you differentiate between the two:

- Most data that exists in text form, including server logs and Facebook posts, is *unstructured*
- Scientific observations, as recorded by careful scientists, are kept in a very neat and organized (*structured*) format
- A genetic sequence of chemical nucleotides (for example, ACGTATTGCA) is *unstructured* even if the order of the nucleotides matters as we cannot form descriptors of the sequence using a row/column format without taking a further look

Structured data is generally thought of as being much easier to work with and analyze. Most statistical and machine learning models were built with structured data in mind and cannot work on the loose interpretation of unstructured data. The natural row and column structure is easy to digest for human and machine eyes. So why even talk about unstructured data? Because it is so common! Most estimates place unstructured data as 80-90% of the world's data. This data exists in many forms and for the most part, goes unnoticed by humans as a potential source of data. Tweets, e-mails, literature, and server logs are generally unstructured forms of data.

While a data scientist likely prefers structured data, they must be able to deal with the world's massive amounts of unstructured data. If 90% of the world's data is unstructured, that implies that about 90% of the world's information is trapped in a difficult format.

So, with most of our data existing in this free-form format, we must turn to pre-analysis techniques, called *preprocessing*, in order to apply structure to at least a part of the data for further analysis. The next chapter will deal with preprocessing in great detail; for now, we will consider the part of preprocessing wherein we attempt to apply transformations to convert unstructured data into a structured counterpart.

Example of data preprocessing

When looking at text data (which is almost always considered unstructured), we have many options to transform the set into a structured format. We may do this by applying new characteristics that describe the data. A few such characteristics are as follows:

- Word/phrase count
- The existence of certain special characters
- The relative length of text
- Picking out topics

I will use the following tweet as a quick example of unstructured data, but you may use any unstructured free-form text that you like, including tweets and Facebook posts.

This Wednesday morn, are you early to rise? Then look East. The Crescent Moon joins Venus & Saturn. Afloat in the dawn skies.

It is important to reiterate that pre-processing is necessary for this tweet because a vast majority of learning algorithms require numerical data (which we will get into after this example).

More than requiring a certain type of data, pre-processing allows us to explore features that have been created from the existing features. For example, we can extract features such as word count and special characters from the mentioned tweet. Now, let's take a look at a few features that we can extract from text.

Word/phrase counts

We may break down a tweet into its word/phrase count. The word *this* appears in the tweet once, as does every other word. We can represent this tweet in a structured format, as follows, thereby converting the unstructured set of words into a row/column format:

	this	**wednesday**	**morn**	**are**	**this wednesday**
Word Count	1	1	1	1	1

Note that to obtain this format we can utilize scikit-learn's `CountVectorizer` that we saw in the previous chapter.

Presence of certain special characters

We may also look at the presence of special characters, such as the question mark and exclamation mark. The appearance of these characters might imply certain ideas about the data that are otherwise difficult to know. For example, the fact that this tweet contains a question mark might strongly imply that this tweet contains a question for the reader. We might append the preceding table with a new column, as shown:

	this	**wednesday**	**morn**	**are**	**this wednesday**	**?**
Word Count	1	1	1	1	1	1

Relative length of text

This tweet is 121 characters long.

```
len("This Wednesday morn, are you early to rise? Then look East. The
Crescent Moon joins Venus & Saturn. Afloat in the dawn skies.")
# get the length of this text (number of characters for a string)

# 121
```

The average tweet, as discovered by analysts, is about 30 characters in length. So, we might impose a new characteristic, called **relative length**, (which is the length of the tweet divided by the average length), telling us the length of this tweet as compared to the average tweet. This tweet is actually 4.03 times longer than the average tweet, as shown:

$$\frac{121}{30} = 4.03$$

We can add yet another column to our table using this method:

	this	**wednesday**	**morn**	**are**	**this wednesday**	**?**	**Relative length**
Word Count	1	1	1	1	1	1	4.03

Picking out topics

We can pick out some topics of the tweet to add as columns. This tweet is about astronomy, so we can add another column, as illustrated:

	this	**wednesday**	**morn**	**are**	**this wednesday**	**?**	**Relative length**	**Topic**
Word Count	1	1	1	1	1	1	4.03	astronomy

And just like that, we can convert a piece of text into structured/organized data ready for use in our models and exploratory analysis.

Topic is the only extracted feature we looked at that is not automatically derivable from the tweet. Looking at word count and tweet length in Python is easy; however, more advanced models (called topic models) are able to derive and predict topics of natural text as well.

Being able to quickly recognize whether your data is structured or unstructured can save hours or even days of work in the future. Once you are able to discern the organization of the data presented to you, the next question is aimed at the individual characteristics of the dataset.

Quantitative versus qualitative data

When you ask a data scientist, "what type of data is this?", they will usually assume that you are asking them whether or not it is mostly quantitative or qualitative. It is likely the most common way of describing the *specific* characteristics of a dataset.

For the most part, when talking about quantitative data, you are *usually* (not always) talking about a structured dataset with a strict row/column structure (because we don't assume unstructured data even *has* any characteristics). All the more reason why the preprocessing step is so important.

These two data types can be defined as follows:

- **Quantitative data**: This data can be described using numbers, and basic mathematical procedures, including addition, are possible on the set.
- **Qualitative data**: This data cannot be described using numbers and basic mathematics. This data is generally thought of as being described using "natural" categories and language.

Example – coffee shop data

Say that we were processing observations of coffee shops in a major city using the following five descriptors (characteristics):

Data: Coffee Shop

- Name of coffee shop
- Revenue (in thousands of dollars)
- Zip code
- Average monthly customers
- Country of coffee origin

Each of these characteristics can be classified as either quantitative or qualitative, and that simple distinction can change everything. Let's take a look at each one:

- Name of coffee shop – Qualitative

  The name of a coffee shop is not expressed as a number and we cannot perform math on the name of the shop.

- Revenue – Quantitative

  How much money a cafe brings in can definitely be described using a number. Also, we can do basic operations such as adding up the revenue for 12 months to get a year's worth of revenue.

- Zip code – Qualitative

  This one is tricky. A zip code is always represented using numbers, but what makes it qualitative is that it does not fit the second part of the definition of quantitative—we cannot perform basic mathematical operations on a zip code. If we add together two zip codes, it is a nonsensical measurement. We don't necessarily get a new zip code and we definitely don't get "double the zip code".

- Average monthly customers – Quantitative

  Again, describing this factor using numbers and addition makes sense. Add up all of your monthly customers and you get your yearly customers.

- Country of coffee origin – Qualitative

  We will assume this is a very small café with coffee from a single origin. This country is described using a name (Ethiopian, Colombian), and not numbers.

A couple of important things to note:

- Even though a zip code is being described using numbers, it is not quantitative. This is because you can't talk about the *sum* of all zip codes or an *average* zip code. These are nonsensical descriptions.
- Pretty much whenever a word is used to describe a characteristic, it is a qualitative factor.

If you are having trouble identifying which is which, basically, when trying to decide whether or not the data is qualitative or quantitative, ask yourself a few basic questions about the data characteristics:

- Can you describe it using numbers?
  - No? It is **qualitative.**
  - Yes? Move on to next question.

- Does it still makes sense after you add them together?
    - No? They are **qualitative.**
    - Yes? You probably have **quantitative** data.

This method will help you classify most, if not all, data into one of these two categories.

The difference between these two categories define the types of questions you may ask about each column. For a quantitative column, you may ask questions such as the following:

- What is the average value?
- Does this quantity increase or decrease over time (if time is a factor)?
- Is there a threshold that if this number grew above or be too low would signal trouble for the company?

For a qualitative column, none of the preceding questions can be answered; however, the following questions *only* apply to qualitative values:

- Which value occurs the most and the least?
- How many unique values are there?
- What are these unique values?

Example – world alcohol consumption data

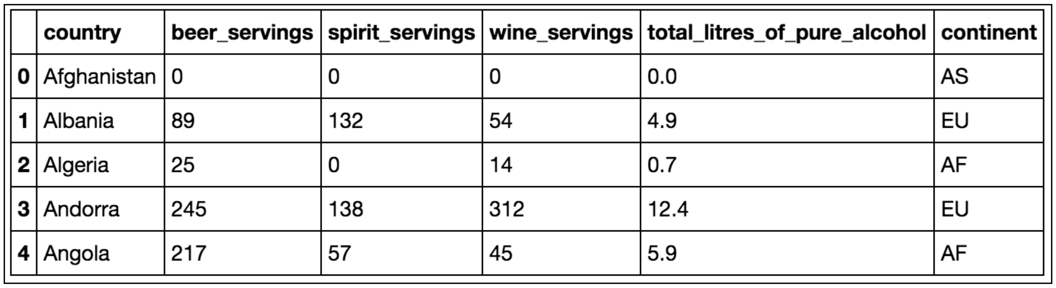
The World Health Organization released a dataset describing the average drinking habits of people in countries across the world. We will use Python and the data exploration tool, Pandas, in order to gain a better look:

```
import pandas as pd

# read in the CSV file from a URL
drinks = pd.read_csv('https://raw.githubusercontent.com/sinanuozdemir/
principles_of_data_science/master/data/chapter_2/drinks.csv')

# examine the data's first five rows
drinks.head()           # print the first 5 rows
```

These three lines have done the following:

- Imported `pandas`, which will be referred to as `pd` in the future
- Read in a **CSV** (**comma separated value**) file as a variable called `drinks`
- Called a method, `head`, that reveals the first five rows of the dataset

Note the neat row/column structure a CSV comes in



	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	AS
1	Albania	89	132	54	4.9	EU
2	Algeria	25	0	14	0.7	AF
3	Andorra	245	138	312	12.4	EU
4	Angola	217	57	45	5.9	AF

We have six different columns that we are working with in this example:

- `country`: Qualitative
- `beer_servings`: Quantitative
- `spirit_servings`: Quantitative
- `wine_servings`: Quantitative
- `total_litres_of_pure_alcohol`: Quantitative
- `continent`: Qualitative

Let's look at the qualitative column `continent`. We can use Pandas in order to get some basic summary statistics about this non-numerical characteristic. The `describe()` method is being used here, which first identifies whether the column is likely quantitative or qualitative and then gives basic information about the column as a whole. This is shown as follows:

```
drinks['continent'].describe()

>> count      193
>> unique       5
>> top         AF
>> freq        53
```

It reveals that the WHO has gathered data about five unique continents, the most frequent being `AF` (Africa), which occurred 53 times in the 193 observations.

If we take a look at one of the quantitative columns and call the same method, we can see the difference in output, as shown:

```
drinks['beer_servings'].describe()

>> mean     106.160622
>> min        0.000000
>> max      376.000000
```

Now we can look at the mean (average) beer serving per-person per-country (106.2 servings) as well as the lowest beer serving, zero, and the highest beer serving recorded, 376 (that's more than a beer a day).

Digging deeper

Quantitative data can be broken down, one step further, into *discrete* and *continuous* quantities.

These can be defined as follows:

- **Discrete data**: This describes data that is counted. It can only take on certain values.

  Examples of discrete quantitative data include a dice roll, because it can only take on six values, and the number of customers in a café, because you can't have a *real* range of people.

- **Continuous data**: This describes data that is measured. It exists on an infinite range of values.

  A good example of continuous data would be a person's weight because it can be 150 pounds or 197.66 pounds (note the decimals). The height of a person or building is a continuous number because an infinite scale of decimals is possible. Other examples of continuous data would be time and temperature.

The road thus far…

So far in this chapter, we have looked at the differences between structured and unstructured data as well as between qualitative and quantitative characteristics. These two simple distinctions can have drastic effects on the analysis that is performed. Allow me to summarize before moving on the second half of the chapter.

Data as a whole can either be *structured* or *unstructured*, meaning that the data can either take on an organized row/column structure with distinct features that describe each row of the dataset, or exist in a free-form state that usually must be preprocessed into a form that is easily digestible.

If data is structured, we can look at each column (feature) of the dataset as being either *quantitative* or *qualitative*. Basically, can the column be described using mathematics and numbers or not? The next part of this chapter will break down data into four very specific and detailed levels. At each order, we will apply more complicated rules of mathematics, and in turn, we can gain a more intuitive and quantifiable understanding of the data.

The four levels of data

It is generally understood that a specific characteristic (feature/column) of structured data can be broken down into one of four levels of data. The levels are:

- The nominal level
- The ordinal level
- The interval level
- The ratio level

As we move down the list, we gain more structure and, therefore, more returns from our analysis. Each level comes with its own accepted practice in measuring the `center` of the data. We usually think of the mean/average as being an acceptable form of center, however, this is only true for a specific type of data.

The nominal level

The first level of data, the *nominal* level, (which also sounds like the word name) consists of data that is described purely by name or category. Basic examples include gender, nationality, species, or yeast strain in a beer. They are not described by numbers and are therefore qualitative. The following are some examples:

- A type of animal is on the nominal level of data. We may also say that if you are a chimpanzee, then you belong to the mammalian class as well.
- A part of speech is also considered on the nominal level of data. The word *she* is a pronoun, and it is also a *noun*.

Of course, being qualitative, we cannot perform any quantitative mathematical operations, such as addition or division. These would not make any sense.

Mathematical operations allowed

We cannot perform mathematics on the nominal level of data except the basic *equality* and *set membership* functions, as shown in the following two examples:

- *Being a tech entrepreneur* is the same as *being in the tech industry*, but not vice versa
- A figure described as a square falls under the description of being a rectangle, but not vice versa

Measures of center

A **measure of center** is a number that describes what the data *tends to*. It is sometimes referred to as the *balance point* of the data. Common examples include the mean, median, and mode.

In order to find the *center* of nominal data, we generally turn to the *mode* (the most common element) of the dataset. For example, look back at the WHO alcohol consumption data. The most common continent surveyed was Africa, making that a possible choice for the *center* of the continent column.

Measures of center such as the mean and median do not make sense at this level as we cannot order the observations or even add them together.

What data is like at the nominal level

Data at the nominal level is mostly categorical in nature. Because we generally can only use words to describe the data, it can be lost in translation among countries, or can even be misspelled.

While data at this level can certainly be useful, we must be careful about what insights we may draw from them. With only the mode as a basic measure of center, we are unable to draw conclusions about an *average* observation. This concept does not exist at this level. It is only at the next level that we may begin to perform true mathematics on our observations.

The ordinal level

The nominal level did not provide us with much flexibility in terms of mathematical operations due to one seemingly unimportant fact—we could not order the observations in any natural way. Data in the *ordinal* level provides us with a rank order, or the means to place one observation before the other; however, it does not provide us with relative differences between observations, meaning that while we may order the observations from first to last, we cannot add or subtract them to get any real meaning.

Examples

The *Likert* is among the most common ordinal level scales. Whenever you are given a survey asking you to rate your satisfaction on a scale from 1 to 10, you are providing data at the ordinal level. Your answer, which must fall between 1 and 10, can be ordered: eight is better than seven while three is worse than nine.

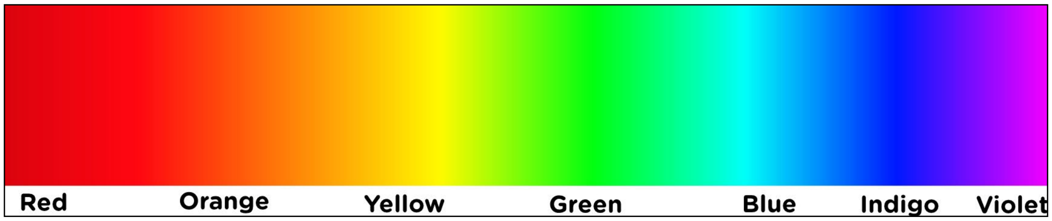
However, differences between the numbers do not make much sense. The difference between a seven and a six might be different than the difference between a two and a one.

Mathematical operations allowed

We are allowed much more freedom on this level in mathematical operations. We inherit all mathematics from the ordinal level (equality and set membership) and we can also add the following to the list of operations allowed in the nominal level:

- Ordering
- Comparison

Ordering refers to the natural order provided to us by the data; however, this can be tricky to figure out sometimes. When speaking about the spectrum of visible light, we can refer to the names of colors—red, orange, yellow, green, blue, indigo, and violet. Naturally, as we move from left to right, the light is gaining energy and other properties. We may refer to this as a natural order.



However, if needed, an artist may impose another order on the data, such as sorting the colors based on the cost of the material to make the said color. This could change the order of the data but as long as we are consistent in what defines the order, it does not matter what defines it.

Comparisons are another new operation allowed at this level. At the ordinal level, it would not make sense to say that one country was *naturally* better than another or that one part of speech is worse than another. At the ordinal level, we can make these comparisons. For example, we can talk about how putting a "7" on a survey is worse than putting a "10".

Measures of center

At the ordinal level, the **median** is usually an appropriate way of defining the center of the data. The mean, however, would be impossible because division is not allowed at this level. We can also use the mode like we could at the nominal level.

We will now look at an example of using the median:

Imagine you have conducted a survey among your employees asking "how happy are you to be working here on a scale from 1-5", and your results are as follows:

```
5, 4, 3, 4, 5, 3, 2, 5, 3, 2, 1, 4, 5, 3, 4, 4, 5, 4, 2, 1, 4, 5, 4,
3, 2, 4, 4, 5, 4, 3, 2, 1
```

Let's use Python to find the median of this data. It is worth noting that most people would argue that the mean of these scores would work just fine. The reason that the mean would not be as mathematically viable is because if we subtract/add two scores, say a score of four minus a score of two, the difference of two does not really mean anything. If addition/subtraction among the scores doesn't make sense, the mean won't make sense either.

```
import numpy

results = [5, 4, 3, 4, 5, 3, 2, 5, 3, 2, 1, 4, 5, 3, 4, 4, 5, 4, 2, 1,
4, 5, 4, 3, 2, 4, 4, 5, 4, 3, 2, 1]


sorted_results = sorted(results)


print sorted_results
'''
[1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 5, 5, 5, 5, 5, 5, 5]
'''


print numpy.mean(results)    # == 3.4375


print numpy.median(results)  # == 4.0
```

The ''' (triple apostrophe) denotes a longer (over two lines) comment. It acts in a way similar to the #.

Turns out that the median is not only more sound, but makes the survey results look much better.

Quick recap and check

So far we have seen half of the levels of data:

- The nominal level
- The ordinal level

At the nominal level, we deal with data usually described using vocabulary (but sometimes with numbers), with no order, and little use of mathematics.

At the ordinal level, we have data that can be described with numbers and also have a "natural" order, allowing us to put one in front of the other.

Let's try to classify the following example as either ordinal or nominal (answers are at the end of the chapter):

- The origin of the beans in your cup of coffee
- The place someone receives after completing a foot race
- The metal used to make the medal that they receive after placing in the said race
- The telephone number of a client
- How many cups of coffee you drink in a day

The interval level

Now we are getting somewhere interesting. At the interval level, we are beginning to look at data that can be expressed through very quantifiable means, and where much more complicated mathematical formulas are allowed. The basic difference between the ordinal level and the interval level is, well, just that—difference.

Data at the interval level allows meaningful subtraction between data points.

Example

Temperature is a great example of data at the interval level. If it is 100 degrees Fahrenheit in Texas and 80 degrees Fahrenheit in Istanbul, Turkey, then Texas is 20 degrees warmer than Istanbul. This simple example allows for so much more manipulation at this level than previous examples.

(Non) Example

It seems as though the example in the ordinal level (using the one to five survey) fits the bill of the interval level. However, remember that the *difference* between the scores (when you subtract them) does not make sense, therefore, this data cannot be called at the interval level.

Mathematical operations allowed

We can use all the operations allowed on the lower levels (ordering, comparisons, and so on), alongwith two other notable operations:

- Addition
- Subtraction

The allowance of these two operations allows us to talk about data at this level in a whole new way.

Measures of center

At this level, we can use the median and mode to describe this data; however, usually the most accurate description of the center of data would be the **arithmetic mean**, more commonly referred to as, simply, "the mean". Recall that the definition of the mean requires us to add together all the measurements. At the previous levels, addition was meaningless; therefore, the mean would have lost extreme value. It is only at the interval level and above that the arithmetic mean makes sense.

We will now look at an example of using the mean.

Suppose we look at the temperature of a fridge containing a pharmaceutical company's new vaccine. We measure the temperate every hour with the following data points (in Fahrenheit):

```
31, 32, 32, 31, 28, 29, 31, 38, 32, 31, 30, 29, 30, 31, 26
```

Using Python again, let's find the mean and median of the data:

```
import numpy


temps = [31, 32, 32, 31, 28, 29, 31, 38, 32, 31, 30, 29, 30, 31, 26]



print numpy.mean(temps)    # == 30.73

print numpy.median(temps)  # == 31.0
```

Note how the mean and median are quite close to each other and both are around 31 degrees. The question, *on average, how cold is the fridge?,* about 31, however the vaccine comes with a warning:

Do not keep this vaccine at a temperature under 29 degrees.

Note that at least twice, the temperature dropped below 29 degrees but you ended up assuming that it isn't enough for it to be detrimental.

This is where the measure of variation can help us understand how bad the fridge situation can be.

Measures of variation

This is something new that we have not yet discussed. It is one thing to talk about the center of the data but, in data science, it is also very important to mention how "spread out" the data is. The measures that describe this phenomenon are called **measures of variation**. You have likely heard of "standard deviation" before and are now experiencing mild PTSD from your statistics classes. This idea is extremely important and I would like to address it briefly.

A measure of variation (like the standard deviation) is a number that attempts to describe how spread out the data is.

Along with a measure of center, a measure of variation can almost entirely describe a dataset with only two numbers.

Standard deviation

Arguably, standard deviation is the most common measure of variation of data at the interval level and beyond. The standard deviation can be thought of as the "average distance a data point is at from the mean". While this description is technically and mathematically incorrect, it is a good way to think about it. The formula for standard deviation can be broken down into the following steps:

1. Find the mean of the data.
2. For each number in the dataset, subtract it from the mean and then square it.
3. Find the average of each square difference.
4. Take the square root of the number obtained in step three. This is the standard deviation.

Notice how, in the steps, we do actually take an arithmetic mean as one of the steps.

For example, look back at the temperature dataset. Let's find the standard deviation of the dataset using Python:

```
import numpy

temps = [31, 32, 32, 31, 28, 29, 31, 38, 32, 31, 30, 29, 30, 31, 26]

mean = numpy.mean(temps)    # == 30.73

squared_differences = []
# empty list o squared differences

for temperature in temps:
    difference = temperature - mean
 # how far is the point from the mean

    squared_difference = difference**2
    # square the difference

    squared_differences.append(squared_difference)
    # add it to our list


average_squared_difference = numpy.mean(squared_differences)
# This number is also called the "Variance"


standard_deviation = numpy.sqrt(average_squared_difference)
# We did it!


print standard_deviation  # == 2.5157
```

All of this code led to us find out that the standard deviation of the dataset is around 2.5, meaning that "on average", a data point is 2.5 degrees off from the average temperature of around 31 degrees, meaning that the temperature could likely dip below 29 degrees again in the near future.

> The reason we want the "square difference" between each point and the mean and not the "actual difference" is because squaring the value actually puts emphasis on outliers—data points that are abnormally far away.

Measures of variation give us a very clear picture of how spread out or dispersed our data is. This is especially important when we are concerned with ranges of data and how data can fluctuate (think percent return on stocks).

The big difference between data at this level and at the next level lies in something that is not obvious.

Data at the interval level does not have a "natural starting point or a natural zero". However, being at zero degrees Celsius does not mean that you have "no temperature".

The ratio level

Finally, we will take a look at the ratio level. After moving through three different levels with differing levels of allowed mathematical operations, the ratio level proves to be the strongest of the four.

Not only can we define order and difference, the ratio level allows us to *multiply and divide* as well. This might seem like not much to make a fuss over but it changes almost everything about the way we view data at this level.

Examples

While Fahrenheit and Celsius are stuck in the interval level, the Kelvin scale of temperature boasts a natural zero. A measurement zero Kelvin literally means the absence of heat. It is a non-arbitrary starting zero. We can actually scientifically say that 200 Kelvin is twice as much heat as 100 Kelvin.

Money in the bank is at the ratio level. You can have "no money in the bank" and it makes sense that $200,000 is "twice as much as" $100,000.

> Many people may argue that Celsius and Fahrenheit also have a starting point (mainly because we can convert from Kelvin to either of the two). The real difference here might seem silly, but because the conversion to Celsius and Fahrenheit make the calculations go into the negative, it does not define a clear and "natural" zero.

Measures of center

The arithmetic mean still holds meaning at this level, as does a new type of mean called the **geometric mean**. This measure is generally not used as much even at the ratio level, but is worth mentioning. It is the square root of the product of all the values.

For example, in our fridge temperature data, we can calculate the geometric mean as shown here:

```
import numpy

temps = [31, 32, 32, 31, 28, 29, 31, 38, 32, 31, 30, 29, 30, 31, 26]

num_items = len(temps)
product = 1.

for temperature in temps:
    product *= temperature

geometric_mean = product**(1./num_items)

print geometric_mean   # == 30.634
```

Note again how it is close to the arithmetic mean and median as calculated before. This is not always the case, and will be talked about at great length in the statistics chapter of this book.

Problems with the ratio level

Even with all of this added functionality at this level, we must generally also make a very large assumption that actually makes the ratio level a bit restrictive.

> Data at the ratio level is usually non-negative.

For this reason alone, many data scientists prefer the interval level to the ratio level. The reason for this restrictive property is because if we allowed negative values, the ratio might not always make sense.

Consider that we allowed debt to occur in our money in the bank example. If we had a balance of $50,000, the following ratio would not really make sense at all:

$$\frac{\$50,000}{-\$50,000} = -1$$

Data is in the eye of the beholder

It is possible to impose structure on data. For example, while I said that you technically cannot use a mean for the one to five data at the ordinal scale, many statisticians would not have a problem using this number as a descriptor of the dataset.

The level at which you are interpreting data is a *huge* assumption that should be made at the beginning of any analysis. If you are looking at data that is generally thought of at the ordinal level and applying tools such as the arithmetic mean and standard deviation, this is something that data scientists must be aware of. This is mainly because if you continue to hold these assumptions as valid in your analysis, you may encounter problems. For example, if you also assume divisibility at the ordinal level by mistake, you are imposing structure where structure may not exist.

Summary

The type of data that you are working with is a very large piece of data science. It must precede most of your analysis because the type of data you have impacts the type of analysis that is even possible!

Whenever you are faced with a new dataset, the first three questions you should ask about it are the following:

- Is the data organized or unorganized?

  For example, does our data exist in a nice, clean row/column structure?

- Is each column quantitative or qualitative?

  For example, are the values numbers, strings, or do they represent quantities?

- At what level of data is each column?

  For example, are the values at the nominal, ordinal, interval, or ratio level?

The answers to these questions will not only impact your knowledge of the data at the end, but will also dictate the next steps of your analysis. They will dictate the types of graphs you are able to use and how you interpret them in your upcoming data models. Sometimes we will have to convert from one level to another in order to gain more perspective. In the coming chapters, we will take a much deeper look at how to deal with and explore data at different levels.

By the end of this book, we will be able to not only recognize data at different levels, but will also know how to deal with it at these levels.