Data preparation 05

Most of us have at some time been abroad in countries where we don't speak the language. With this basic method of communication closed off to us, it can be very difficult to get our meaning across. Even if we do have some knowledge of the language, the gaps in our vocabulary and grammar will often cause us (and the listener) some frustration.

Language, then, is a fundamental necessity if we want to understand and communicate with another individual. And preparing data is all about establishing a common language between human and machine.

In this chapter, we will learn why data should never be analysed without first having been prepared, the step-by-step process in which data can be prepared and the best methods that I have learned for managing problems in datasets.

Encouraging data to talk

As practitioners, unless we are very lucky, data will often come to us 'dirty'. It is often collected by people who do not standardize their records, managed by people who might tamper with their datasets' column and row names to suit their own projects, and stored in non-optimal locations that can cause damage to the data. With so many different people working on a single dataset and using different methods for adding data, the resulting datasets in many organizations are unsurprisingly riddled with errors and gaps. And we cannot expect a machine to know where the errors lie or how to fix inconsistencies in information.

So it is our job to prepare the data in such a way that it can be comprehended – and correctly analysed – by a machine.

With great power comes great responsibility

Data preparation (also known as 'data wrangling') is a complex component of the entire process and, as it comprises a number of tasks that can only be completed manually, this stage normally takes the most amount of time.[1] The reason behind this close attention to data preparation is that, if the raw data is not first structured properly in the dataset, then the later stages of the process will either not work at all or, even worse, will give us inaccurate predictions and/or incorrect results. This can spell disaster for you and your company, and at the very worst end of the scale, neglect of this stage can result in firings and, in the case of freelance work, even lawsuits.

It is not my intention to scare you – I simply mean to show you how essential it is to prepare our data. Surprisingly, despite its importance, this is where I have found a serious gap in the educational materials on data science, which largely focus on the later stages of the process: analysing and visualizing. These books and courses use datasets that have already been prepared. But while this approach may be fine if you're just getting to grips with the discipline, paying no attention to preparation means that you are effectively only learning the cosmetic ways to work with data.

Working only with datasets from educational courses will merely show you data that has been cleaned up to fit the case example. But in the real world, data is often dirty, messy and corrupt, and without knowing the causes and symptoms of dirty data, we cannot adequately complete our project. If you do not prepare your data, when you get out into the real world on your first project, and your algorithm inevitably returns 'missing data' errors, or 'text qualifier' errors, or 'division by zero', your project will grind to a halt.

How do we know, then, that our data has been sufficiently prepared? Quite simply, once we have ensured that it is suitable for our data analysis stage. It must:

- be in the right format;
- be free from errors; and
- have all gaps and anomalies accounted for.

A common phrase that data scientists use is 'garbage in, garbage out', which means if you put unclean data into an algorithm, you will only get nonsensical results, making your analyses useless. It is true that a number of practitioners struggle with this step, but that is only because they don't have a framework to follow. This usually leads to an unstructured,

undocumented approach and means they have to reinvent the wheel every time they prepare their data; an ineffective and time-consuming approach in the long run.

So, let's get started with the process of preparation.

CASE STUDY Ubisoft – making the case for data preparation

Ulf Morys is Finance Director at the German branch of Ubisoft, a game design, development and distribution company that has created popular game franchises from *Assassin's Creed* to *Far Cry*. Ulf oversees a distribution subsidiary, which distributes Ubisoft's video games in the area of Germany, Switzerland and Austria (GSA) and is also responsible for the financial components of the company's operations in central Europe.

Ubisoft had historically had its data used solely by the production team for in-game analytics and monetization. Until Ulf changed matters, finance did not number among its strategic areas for data science.[2] But ignoring the improvements that data science makes can be a costly oversight, and having prior experience of leveraging data to make important business decisions (in his previous job, Ulf saved his company $40 million in a merger thanks to his attention to the data) meant that he knew that a deliberate strategy for using company data was essential.

He says, crucially:

> Preparing data doesn't add more data, it just improves the way you look at it. It's like that scene in the film *The Wizard of Oz* when Dorothy opens the door of her house to the kingdom of Oz and the black and white world of Kansas is replaced with Technicolor. There's nothing fundamentally different to the way the story functions on a technical level, and yet everything has changed. It has been cleaned up.
>
> (SuperDataScience, 2016)

To better understand how Ubisoft in particular benefited from data preparation, Ulf turned to Ubisoft's production team, who had been collecting data from its thousands of online gamers for years, from the length of time the game was played and the length of time that it took to complete individual levels, to what the player did in the game and where they failed on the game map. Ulf found that they were using this past data to assess the likelihood that customers would purchase in-game products through a 'freemium model'.[3] Having the data to hand

not only helped Ubisoft to find out the purchasing patterns of its core customers but also to identify the behaviours that could be applied to future players.

Talking to his team about the grand steps the production team were making thanks to data science turned their heads to the idea. During a financial strategy meeting, Ulf's team mapped out the sources of all Ubisoft's available data and what was missing from the picture, which offered something tangible from which colleagues could bounce off their ideas. 'Very simply,' says Ulf, 'if you don't know something is there, you can't ask questions of it' (SuperDataScience, 2016).

The gaps showed what key data they needed to gather from their customers (store dimensions, space provided for selling video games, attitudes of typical buyers to video games and consumer feelings about Ubisoft's output) before they could make meaningful analyses. Ulf says:

> It was evident why we weren't taking a more systematic approach to our customers: *we didn't have the data*. When I went to our sales department, I was told which of our customers were good based on their own insider knowledge. But that knowledge hadn't been collected systematically. Getting the data – the hard facts – was absolutely necessary.
>
> (SuperDataScience, 2016)

Gathering that information for 2,000 stores enabled Ulf to prepare statistically relevant data that would finally be suitable for analysis. This helped Ubisoft to better target its customers in a way that had not been possible before.

Preparing your data for a journey

In order to get our raw data (information that has not been prepared) to respond to analysis, we first have to prepare it. The method for doing so comprises just three stages:

1. **Extract** the data from its sources;
2. **Transform** the data into a comprehensible language for access in a relational database;
3. **Load** the data into the end source.

This process is known as ETL, and it will help us to gather our data into a suitable format in our end source – known as the 'warehouse' – which can be accessed and analysed in the later stages of the Data Science Process. A data warehouse stores otherwise disparate data in a single system. Oftentimes, it will comprise relational databases.

What is a relational database?

Relational databases allow us to examine relational data across them. In this type of database, the relationships *between* the units of information across datasets matter.

The datasets in a relational database are linked by columns that share the same name. For example, if multiple datasets contained columns with the header 'Country', the data from those columns could be compared across them in a relational database. The benefit of having this type of database is that they facilitate the methods of analysis and visualization that are required to derive insights, where data can be examined across multiple sets without the need for individual extraction.

Perhaps the best way to illustrate how advantageous a relational database can be is to compare it to Excel, which is frequently used by people unaccustomed to working with databases:

1. **It maintains integrity.** Every cell in Excel is individual; there are no limitations to the types of values that you can place into them. You can add dates or text, for example, underneath phone numbers or monetary values, and Excel will be perfectly happy. A relational database will rap you over the wrist for such negligence. In a database, columns have predefined types, which means that a column that has been set up to accept dates will not accept any value that does not fit the date format. Databases, then, will keep an eye on the process for you, querying any values that do not match those predefined by the column.
2. **It combines datasets.** Combining datasets within a relational database is easy; it is much harder to do in Excel. Relational databases have been designed for that purpose, and that makes it easy to create new datasets from combining common values across the relational database. All that is required of you is the ability to execute a simple command. As combining tables is not a primary function of Excel,[4] an advanced knowledge of programming is required to shoehorn your data into a single table.
3. **It is scalable.** Relational databases have been especially designed for scalability; as they combine datasets, it is expected that they must be able to cope with a large number of informational units. That means – regardless of whether you have five or five billion rows – your relational database is unlikely to crash at a crucial moment. Excel is far more limited in this capacity, and as your dataset grows, the software's performance will deteriorate as it struggles to cope with the overload.

The data cleanse

We know that in the real world, data is more likely to come to us dirty, but there is some disagreement among practitioners as to how and when data should be cleaned. Some people clean before they transform, and others only once they have loaded it into the new database. My preference is to clean the data at *each stage* of the ETL process – it might seem an inefficient use of your time but I have found this to be the best way to protect yourself against obstacles further on. Unfortunately, data preparation is always going to be time-consuming, but the more due diligence you take in this stage, the more you will speed up the Data Science Process as a whole.

1 Extract your data

We need to extract data in the first instance 1) to ensure that we are not altering the original source in any way, and 2) because the data that we want to analyse is often stored across a number of different locations. Some examples of possible locations are:

- a database;
- an Excel spreadsheet;
- a website;
- Twitter;
- a .csv file;
- paper reports.

If we are using data from multiple sources, then we will have to extract it into a single database or warehouse in order for our analyses to work. But it is not always easy to extract from locations that use formatting particular to that system – Excel is one such culprit, to which we will return later on in this chapter.

.csv files

You will get to know these types of files quite intimately as a data scientist. They are the simplest type of raw files of data – completely stripped of any formatting – which makes them accessible to any number of programs into which we may want to import them. In .csv files, rows are placed on

new lines, and columns are separated by commas in each line. Hence the abbreviation, which stands for 'comma separated values'.

The beauty of working with raw files is that you will never lose or corrupt information when you load your dataset into a program. This is why they are the standard for most practitioners.

Why it's important to extract data even if it is only in one location

Technically, you *could* analyse the data directly within its storage facility (the original database, an Excel spreadsheet and so on). While it is not recommended, this method is acceptable for making quick calculations, such as computing the sum of a column of values in Excel. However, for serious data science projects, carrying out data tasks within its original storage facility is a huge red flag. In doing so, you might accidentally modify the raw data, thereby jeopardizing your work.

And this is the *best-case* scenario, as it only affects you and your individual project. Working within the storage facility rather than extracting the original data to a test database leaves it vulnerable to user damage, and your work may even end up crashing the internal systems of your institution. That should give any data scientist pause when they start working with an organization's data. They are entrusting us with important if not essential company information, so we must ensure that we leave the data just as it was when we started on the project.

Software for extracting data

There are a couple of exceptional free-to-use programs for extracting and reading data that are sure to wean you off any bad habits as have often been formed by Excel users. These programs work well with data that is in a raw .csv file format.[5]

Although it can take time, data can in most cases be stripped down to a raw .csv file. And if you're working for a large organization where you have to request data extracts, then good news: the data will most likely be given to you in a .csv format anyway.

Notepad++ This is my go-to tool when I want to look at the data I have extracted. Among other features it is a powerful editor for viewing .csv files, and it is much more user-friendly than the notepad software that comes

as standard with Windows. Notepad++ also has a few other significant advantages:

- row numbering, enabling you to navigate through your files and keep tabs on where potential errors might be found;
- a search and replace feature, which enables you to quickly find values or text that you don't want in the dataset and amend them;
- it has been designed for purpose, which means that you can be confident it will not inadvertently modify your data as will other spreadsheet software;
- while the Notepad software that comes with Windows generally has trouble dealing with large files, Notepad++ can open files up to 2 GB.

EditPad Lite EditPad Lite is a program that is free for personal use. It offers similar features to Notepad++, with one major benefit: although both work well with files that are under 2 GB, I have noticed that Notepad++ can sometimes struggle with datasets at the top end of this file size. As a result, I have found EditPad Lite to perform much better with my larger files. If you find that you are overworking Notepad++ with your files, consider EditPad Lite.

2 Transform your data

You cannot simply dump your data from its original source directly into a data warehouse. Not unless you *want* to work with a messy dataset. By transforming your data, you can reformat the information you plan to use into a language that will suit your objectives.

In a broad sense, the transformation step includes alterations such as joining, splitting and aggregating data. These are functions that allow us to create derived tables to better suit the problem at hand. But the most important function of transformation is data cleaning – and that's what we will focus on.

In this step, we must identify and manage any errors in our original database, which in the real world will often run the gamut from formatting inconsistencies, through outliers, to significant gaps in information. But to do so, we first have to understand what we are looking for. So, how can we identify dirty data?

Dirty data

Dirty data is information that is either incorrect, corrupt or missing. These three qualifiers are due to the following factors.

Incorrect data In these instances, information has been (partially or completely) incorrectly added to the database (eg inputting a currency value into a date cell). Sometimes, we will know that data is incorrect. It may be evident when there is a mismatch between columns.

For example, if we had a single row, where the country cell was 'France' and the city cell was 'Rome', we would know that one was incorrect. We may also be able to identify incorrect data by simply using our common sense – we would know that an entry in a date of birth column that has been given as '12/41/2001' simply cannot be correct.

Corrupt data Corrupt data refers to information that may originally have been correct in the dataset but is now mangled. Information can become corrupted in different ways. Contributing factors can include if the database to which it belongs has been physically damaged, if it has been altered by another software or if it has been previously extracted in unadvisable ways. Sometimes, data can simply become corrupted due to transfer to a database that does not support the format it had in the previous storage.

Missing data Missing data either occurs when no information is available for a given cell, or when the person responsible for inserting the data has neglected to add it into the cell. Missing data is a common topic in data science, and it is most likely to occur because of human error.

What can happen when we don't deal with missing data

We should always be aware of any gaps in our information. Below, you'll see a real-life example of data that we have extracted from an Excel spreadsheet into a .csv file that shows dividend pay-outs, organized by year.

5.1

```
487 19-May-15,533.98,540.66,533.04,537.36,537.36,1966900
488 18-May-15,532.01,534.82,528.85,532.3,532.3,2003400
489 15-May-15,539.18,539.27,530.38,533.85,533.85,1971300
490 14-May-15,533.77,539,532.41,538.4,538.4,1403900
491 13-May-15,530.56,534.32,528.66,529.62,529.62,1252300
492 12-May-15,531.6,533.21,525.26,529.04,529.04,1634200
493 11-May-15,538.37,541.98,535.4,535.7,535.7,905300
494 08-May-15,536.65,541.15,525,538.22,538.22,1527600
495 07-May-15,523.99,533.46,521.75,530.7,530.7,1546300
496 06-May-15,531.24,532.38,521.09,524.22,524.22,1567000
497 05-May-15,538.21,539.74,530.39,530.8,530.8,1383100
498 04-May-15,538.53,544.07,535.06,540.78,540.78,1308000
499 01-May-15,538.43,539.54,532.1,537.9,537.9,1768200
500 30-Apr-15,547.87,548.59,535.05,537.34,537.34,2082200
501 29-Apr-15,550.47,553.68,546.91,549.08,549.08,1698800
502 28-Apr-15,554.64,556.02,550.37,553.68,553.68,1491000
503 27-Apr-15,563.39,565.95,553.2,555.37,555.37,2398000
504 26-Apr-15,10000000/10000000 Stock Split,,,,
505 24-Apr-15,564.55,569.58,555.72,563.51,563.51,4932500
506 23-Apr-15,539.52,549.45,538.75,545.5,545.5,4184800
507 22-Apr-15,532.94,539.6,530.29,537.89,537.89,1593500
508 21-Apr-15,536.04,537.91,532.21,532.51,532.51,1844700
509 20-Apr-15,524.16,534.62,523.06,533.91,533.91,1679200
510 17-Apr-15,527.21,528.39,519.58,522.62,522.62,2151800
511 16-Apr-15,528.45,534.12,528.16,532.34,532.34,1299800
512 15-Apr-15,527.25,533.27,521.79,531.07,531.07,2318800
513 14-Apr-15,534.78,536.1,526.65,528.94,528.94,2604100
```

As you can see from the commas enclosing no information, five of the columns at the highlighted row 504 (26-Apr-15) have missing fields of data.

We have been lucky in this instance that the missing columns have survived the extraction – oftentimes, the missing data values are not defined by commas. What this would mean is that when we plug the dataset into an algorithm, it would recalibrate our data incorrectly, pushing the data in the row below up to fit the number of columns required in the dataset. In the above example, this would mean that the date 24-Apr-15 would be brought up to the column directly to the right of the '10000000/10000000 Stock Split' value.

Missing data in this way can cause us significant trouble in the analysis stage if we don't catch the problem beforehand. I have known some newbie data scientists who will check the top 100 rows of their dataset, but this is a rookie mistake – if there are errors in the data, you are much more likely to see them at the end of the dataset because the errors will shift information.

Fixing corrupt data

To fix corrupt data so that it can be read by a machine, we can first try the following:

- re-extract it from its original file to see if something has corrupted the file during the first extraction;

- talk to the person in charge of the data to see if they can cast light on what the actual data should be; or
- exclude the rows that contain corrupt data from your analysis.[6]

Approaching people

If you find yourself in a situation where you are missing data and need to retrace your steps to obtain additional input before the project can progress, here are three ways in which you can facilitate the process:

- Always be courteous to the people who are giving you data. Some people may find data collection frustrating and will let that show in their communication, but try to stay neutral. Remember that they are not data scientists, and may not take the same joy in the process of gathering data as you! Explain to them that every data-driven project will have different outcomes, and that each project requires different types of data. You may need to approach the team responsible for your datasets multiple times, so be friendly and get them on your side.
- Make sure that anyone you speak to fully understands the problem that you are trying to solve, as well as their role in combating it. Seeing the bigger picture will help your colleagues be more patient with your requests.
- Keep a list of the company's data assets with you at all times. Having this to hand means that when you're off on the hunt for new data, you will be able to cross-check what the organization already has and reduce the likelihood of you collecting duplicates. When you list the data assets, I recommend recording the names of the data sources as well as the databases' columns and their descriptors.

Fixing missing data

If we cannot resolve our problem by using any one of these methods, then we must consider our data as missing. There are various methods for resolving the problem of missing fields in spreadsheets:

- **Predict the missing data with 100 per cent accuracy.** We can do this for information that we can derive from other data. For example, say we have a spreadsheet with customer location data that contains column values for both 'State' and 'City'; the entry for State is missing but the

City entry is 'Salt Lake City'. Then we can be certain that the state is 'Utah'.[7] It is also possible to derive a missing value based on more than one value, for example, to derive a profit value from both revenue and expenses values. Bear in mind that when we are inputting information in both examples, we are doing so on the assumption that there were no errors in the collection of the data.

- **Leave the record as it is.** In this scenario, you would simply leave the cell with no data empty. This is most useful when specific fields have no bearing on our analysis and therefore can be left out of our testing, but it can also be used if we are planning to use a method that isn't significantly affected by missing data (ie methods that can use averaged values) or if we use a software package that can deal appropriately with this lack of information. In cases where you leave the record as it is, I would recommend keeping notes of where your data contains gaps, so that any later anomalies can be accounted for.
- **Remove the record entirely.** Sometimes, the data that is missing would have been critical to our analysis. In these instances, our only option is to remove the entire row of data from our analysis, as the missing information makes them unable to contribute. Obviously, the major drawback in this case is that our results will become less significant as the sample has decreased. So this approach is likely to work best with large datasets, where the omission of a single row will not greatly affect the dataset's statistical significance.
- **Replace the missing data with the mean/median value.** This is a popular approach for columns that contain numerical information, as it allows us to arbitrarily fill any gaps without tampering too significantly with our dataset. To calculate the mean, we add all of the values together and divide that total by the number of values. To calculate the median, we find the sequential middle value in our data range (if there are an uneven number of values, just add the two middle numbers and divide that total by two). Calculating the median rather than the mean is usually preferable, because the former is less affected by outliers, which means that extreme values either side of the median range will not skew our results.
- **Fill in by exploring correlations and similarities.** This approach is again dependent on your missing data value being numerical, and it requires the use of models to predict what the missing values might have been. For instance, we could use a predictive algorithm (such as K-nearest neighbours, which we will discuss in Chapter 6) to forecast the missing data based on existing similarities among records in your dataset.

- **Introduce a dummy variable for missing data.** This requires adding a column to our dataset: wherever we find missing values in the dataset, we allocate a 'yes' value to it – and when it is not missing we give it a 'no' value. We can then explore how the variable correlates with other values in our analysis, and so retrospectively consider the implications of why this data might be missing.

Dealing with outliers

Let's say that we are working for a company selling phone accessories and we want to find the average number of units that we have sold of one of our phone cases to each of our distributors. We have been in business for years, and so our datasets are large. The person responsible for inputting these values into our database was having a bad day, and instead of inputting the number of product units into the product column, they put the distributor's telephone number. That error would abnormally raise our average in this column (and would mean that a single distributor has purchased at least 100 million units!).

If we were to analyse that record on its own, we would probably notice the error. But if we simply calculated the average without looking at the data, our report would be skewed by that outlier – and that would make the report unusable.

Nevertheless, it's important to distinguish between outliers that can be attributed to erroneous information and outliers that are correct but that fall outside the normal range of values. The value for a distributor that *did* purchase 100 million units of your product will still be an outlier, as the value is higher than the normative number of units purchased.

Many datasets will have outliers – our job is to understand where they are and to ensure that they do not unfairly skew our reports. This will largely depend on the type of analysis that we want to carry out. For example, if we wanted to work out for a publishing house the average number of units sold to book stores around the world, and we know that the outlier was an exceptional purchase order, we might choose to remove the record even though it's valid.

It is possible to find outliers in your dataset without searching for them manually, by generating a distribution curve (also known as a bell curve for normal distributions) from your column values. Distribution curves graphically depict the most probable value or event from your data by way of their apex, and it is simple enough to create them directly, even in Excel.[8] Once you have created your distribution curve, you can identify the values that fall outside the normal range.

CASE STUDY Applied approaches to dealing with dirty data

We have been given a dataset from an imaginary venture capital fund that is looking at the overall growth of start-ups in the United States. As the data gatherer was not affiliated with the start-ups, some information was missing, as it was either not publicly available or the start-ups were unwilling to provide that level of information.

5.2

ID	Name	Industry	Inception	Employees	State	City	Revenue	Expenses	Profit	Growth
1	Over-Hex	Software	2006	25	TN	Franklin	$9,684,527	1,130,700 Dollars	8553827	19%
2	Unimattax	IT Services	2009	38		Kirill:	$14,016,543	804,035 Dollars	13212508	20%
3	Greenfax	Retail	2012			Proxy with industry Median	$9,746,272	1,044,375 Dollars	8701897	16%
4	Blacklane	IT Services	2011	66			$15,359,369	4,631,808 Dollars	10727561	19%
5	Yearflex	Software	2013	45			$8,567,910	4,374,841 Dollars	4193069	19%
6	Indigoplanet	IT Services	2013	60			$12,805,452	4,626,275 Dollars	8179177	22%
7	Treslam	Financial Services	2009	116	MO	Clayton	$5,387,499	2,127,984 Dollars	3259485	17%
8	Rednimdox	Construction	2013	73	NY	Woodside				
9	Lamtone	IT Services	2009	55	CA	San Ramon	$11,757,018	6,482,465 Dollars	5274553	30%
10	Stripfind	Financial Services	2010	25	FL	Boca Raton	$12,329,371	916,455 Dollars	11412916	20%
11	Canecorporation	Health	2012	6		New York	$10,597,009	7,591,189 Dollars	3005820	7%
12	Mattouch	IT Services	2013	6	WA	Bellevue	$14,026,934	7,429,377 Dollars	6597557	26%
13	Techdrill	Health	2009	9	MS	Flowood	$10,573,990	7,435,363 Dollars	3138627	8%
14	Techline		2006	65	CA	San Ramon	$13,898,119	5,470,303 Dollars	8427816	23%
15	Cityace		2010	25	CO	Louisville	$9,254,614	6,249,498 Dollars	3005116	6%
16	Kayelectronics	Health	2009	687	NC	Clayton	$9,451,943	3,878,113 Dollars	5573830	4%
17	Ganzlax	IT Services	2011	75	NJ	Iselin	$14,001,180		11901180	18%
18	Trantraxlax	Government Services	2011	35	VA	Suffolk	$11,088,336	5,635,276 Dollars	5453060	7%
19	E-Zim	Retail	2008	320	OH	Monroe	$10,746,451	4,762,319 Dollars	5984132	13%
20	Daltfase	Software	2011	78	NC	Durham	$10,410,628	6,196,409 Dollars	4214219	17%
21	Hotlane	Government Services	2012	87	AL	Huntsville	$7,976,332	5,686,574 Dollars	2291758	2%
22	Lathotline	Health		103	VA	McLean	$9,418,303	7,567,233 Dollars	1851070	2%
23	Lambam	IT Services	2012	210	SC	Columbia	$11,950,148	4,365,512 Dollars	7584636	20%
24	Quozap	Software	2004	21	NJ	Collingswood	$8,304,480	7,019,973 Dollars	1284507	20%
25	Tampware	Construction	2011	13	TX	Houston	$9,785,982	2,910,756 Dollars	6875226	11%
26	Dalthow	Health	2000	20	GA	Dacula	$10,800,718	7,731,820 Dollars	3068898	7%
27	Ranktech	Government Services	2010	607	FL	Tampa	$10,515,557	7,439,384 Dollars	3076173	8%
28	Unadex	Software	2013	280	NC	Cary	$5,231,275	2,388,521 Dollars	2842754	19%

As you can see, various types of information are missing across our columns, and sometimes multiple values are empty in a single row. Let's put the methods for fixing missing data into practice. Return to the methods given above and consider how you might resolve the problem of missing data yourself before reading the answers below.

Employees

Replace the missing data with the mean/median value. This is a numerical value, and so we can proxy any of the missing employee values with the overall or industry median for that column. (The industry median is preferable as it will provide a like-for-like figure.)

Industry

Leave the record as it is *or* predict the missing data with 100 per cent accuracy *or* remove the record entirely. It should be relatively easy to find out to which industry the company belongs by simply investigating what it does and taking your cues from there. But our choice depends on how important industry is to

our analysis. If industry is important, and we cannot research it, we must remove the record from the analysis.

Inception

Leave the record as it is *or* predict the missing data with 100 per cent accuracy *or* remove the record entirely. Even though inception is a number, it is not a numerical value (you cannot perform arithmetic operations with it). For that reason, we cannot proxy it with an average, and so if we cannot find out when the company was established, then we must accept it as missing.

State

Leave the record as it is *or* predict the missing data with 100 per cent accuracy *or* remove the record entirely. In the example given above, we can predict the missing data with 100 per cent certainty. But we must be careful that we are being accurate: for values where a city name can belong to more than one state, we will not be able to predict the data with 100 per cent accuracy and so must decide how important the data is to our analysis.

Expenses

Predict the missing data with 100 per cent accuracy. This is an easy one; we can calculate expenses by simply subtracting profit from revenue.

Revenue, expenses and profit, growth

Replace the missing data with the mean/median value. Calculating this block of missing values requires taking more than one step. We need to first proxy our growth revenue and expenses by using the industries' medians, and then we can calculate the profit as the difference between revenue and expenses.

Transforming data from MS Excel

Excel tries to make things easier by automatically reformatting certain values. This can lead to various hiccups during the ETL process, and as this program is so frequently used to store data, I will give special attention to it here. One common complaint I have heard from Excel users is the program's insistence on converting long numerical values (such as phone and credit card numbers) into a scientific formula.[9] And that's not

the worst of it. Excel can also convert dates and monetary amounts into a single format that accords to your computer's regional settings. While this may be convenient for single spreadsheets which are often used for business intelligence, those kinds of automations will only end up doing you a disfavour in data science, as Excel's formatting does not translate well into a database. And if we are dealing with a lot of data, unpicking all of the instances that Excel has altered can be time-consuming.

If we do not transform the data from Excel into a .csv file, then we will only be presented with problems further down the line. While it may be possible to restore dates that have been altered, it is near impossible to restore credit card numbers if they have been changed to scientific formulae. Just imagine the consequences for an organization that loses its customers' credit card numbers, especially if you had been working on the only copy of the file.

Some of the most common issues are to do with dates and currency, as these values are not international and are therefore susceptible to our machines' regional settings.

Date formats The formatting of dates will differ depending on our geographic region, and Excel has been programmed to display the date that accords to our computer's regional settings. Most countries use a little-endian date format that begins with the day, followed by the month and the year (dd/mm/yyyy). In the United States, however, the date format begins with the month, followed by the day and year (mm/dd/yyyy). We need to ensure that we are working with a consistent date format in our database.

How to fix them The best method to prevent Excel from making changes to our records is to change all of our date formats to yyyy-mm-dd, as this is the unambiguous international standard that is also not subject to regional rules. In Excel, select the column that you want to fix, right-click, and select 'Format Cells'. In the Category window, select 'Date'. In the 'Type' window you should see the yyyy-mm-dd format. Select that and then click 'OK'. Your dates will have been reformatted.

Currency formats Currency will also depend on our computer's regional settings. In these cases, it is not only necessary to consider the symbol of currency but also the decimal marks that are being used. Symbols of currency should be completely stripped from your data as they will otherwise be read as text. Countries use different decimal marks for their currency – indicated either by a point (eg £30.00 in the UK) or a comma (eg €30,00 in Germany).

Take note that this affects both the decimal place *and* the thousands separator. The sum of £30,000 would be read as thirty thousand pounds in countries such as Australia that use the comma to indicate thousands, but it may be read as thirty pounds in countries such as Sweden that use the comma to indicate decimal places. Databases function with the decimal point system, and any commas, including thousands separators, must be stripped from your data.

How to fix them We want to strip our numbers of symbols and commas. If your country uses a decimal comma system, you must first change the regional settings of your computer to make sure the comma is changed to a dot. Select the column, right-click it and select 'Format Cells'. In the 'Category' window, select 'Currency'. Uncheck the 'Use 1000 separator' box to ensure that no commas will be used, choose 'None' from the 'Symbol' dropdown box and select '2' as the number of decimal places. That will remove the commas and symbols from our data.

3 Load your data

Once we have transformed our data into the format we need, we can load our data into our end target: the warehouse. Once this process is complete, we should manually look through our data one last time before we run it through a machine algorithm, to be absolutely certain that we are not working with underprepared data.

Quality assurance after the load

Loading the data into a warehouse can sometimes cause problems. You may have missed cleaning up some of the dirty data in the previous stage, or some of the data may have simply been loaded incorrectly. For that reason, you must learn to double-check your data within the warehouse.

The following are the quality assurance (QA) checks that you should always make at this stage:

- **Count the number of rows** that you have in your final dataset and compare it to the initial dataset. If it is different, return to the initial dataset to find out what happened. Unfortunately, sometimes the quickest way to check is just by looking at it, and this will mean scrolling through the data line by line. The quickest way to do this is to go from the bottom up rather than the top down, because any errors in data are likely to carry downwards.

- **Check the columns for skewness.** To completely safeguard yourself against problems in the analysis stage, check both the top 100 and the bottom 100 rows.
- **Check the columns that are susceptible to corruption.** This usually refers to dates and balances, as we have earlier established that they are the most prone to error.
- **Check text values.** If we have free-form text values from surveys where respondents have typed up answers to an open-ended question, then uploading this kind of text to a database can be tricky. Usually, databases will limit the maximum number of letters in a column. That might result in cutting off an answer, which leaves our data as missing or can even sometimes affect the rest of the dataset. Free-form text can also contain symbols that databases either cannot recognize or misuse because they are qualifier symbols such as quotation marks.

Think (again) like a consultant

Quality assurance is the most important part of data preparation and, as it comes right at the end of the process, be careful not to lose steam at this stage. I was lucky to enter into the field of data science through the world of consulting, which pays great diligence to QA. With quality assurance, work is peer reviewed. The numbers need to add up, and results have to make sense. Don't be afraid of this stage – it's not designed to trip you up, it is there to help protect you from making errors later on in the process.

Those companies that have worked with data for a while have set up rigorous, predetermined procedures that data scientists must follow to the letter before any analysis can be carried out. Some companies will even have consultants to check your process, and expect them to take a lot of time with this step. Delivering an incorrect result will at the very least cost money and, in the worst case, may severely affect business operations. That is why it is so important to ensure that QA is carried out before you move on to the next step.

Now that you have a beautiful warehouse of squeaky-clean data, and you know the question or series of questions that you want to pose to it, you can finally move on to my favourite part: the analysis.

Reference

SuperDataScience (2016) SDS 008: data science in computer games, learning to learn and a 40m euro case study with Ulf Morys [Podcast] 28 October [Online] www.superdatascience.com/8 [accessed 05.06.17]

Notes

1. Opinions differ among data scientists, but most will attribute 60–80 per cent of project time to the data preparation stage.
2. Many large organizations that have been collecting data for years suffer from an institutional blindness to data science – without knowing that data must be prepared before it can be analysed, their information is unusable.
3. A game that is downloaded for free but that sells in-game items for players who want to advance more quickly in the game.
4. Tabs are used for different tables, but it can be tricky to combine values across them.
5. As you progress in your career as a data scientist you will learn to work with various data storage facilities. We are talking about .csv files here because they are the most common and versatile and are a good place to get started.
6. Your decision will ultimately depend on whether or not you need the data, and that can easily be answered if you have taken the time to identify the question in Stage 1 of the Data Science Process.
7. Be careful with fields like this. There is only one Salt Lake City in the United States, but sometimes you will find more than one city with the same name.
8. MS Office versions vary. Typing 'distribution curve' into Excel's Help menu will return the results you need to generate a curve.
9. For example, 4556919574658621 would be shown as 4.55692E+15.