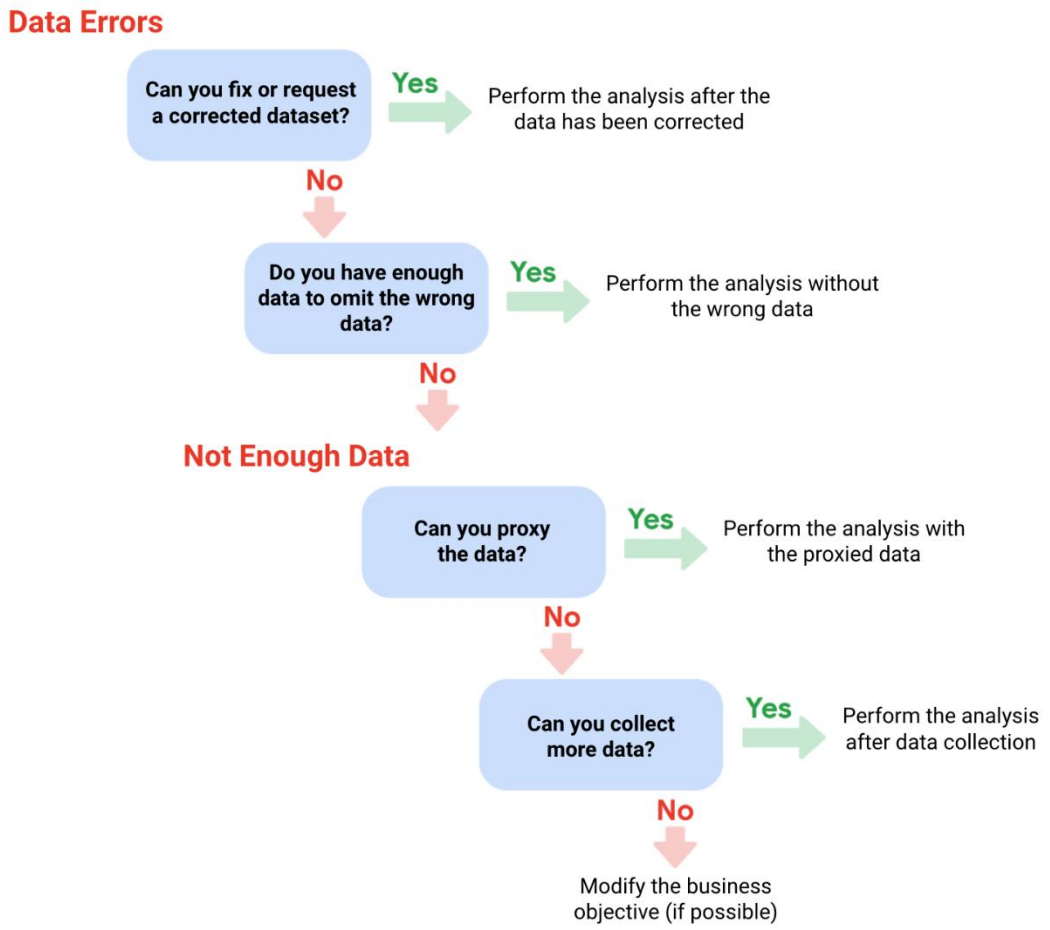
Process

Use the following decision tree as a reminder of how to deal with data errors or not enough data:



1. Can you fix or request a corrected dataset? NO
2. Do you have enough data to omit the wrong data?
NO 3. Can you proxy the data?
NO 4. Can you collect more data?
NO Modify the business objective (if possible)

Calculate sample size

Before you dig deeper into sample size, familiarize yourself with these terms and definitions:

Terminology	Definitions
Population	The entire group that you are interested in for your study. For example, if you are surveying people in your company, the population would be all the employees in your company.
Sample	A subset of your population. Just like a food sample, it is called a sample because it is only a taste. So if your company is too large to survey every individual, you can survey a representative sample of your population.
Margin of error	Since a sample is used to represent a population, the sample's results are expected to differ from what the result would have been if you had surveyed the entire population. This difference is called the margin of error. The smaller the margin of error, the closer the results of the sample are to what the result would have been if you had surveyed the entire population.
Confidence level	How confident you are in the survey results. For example, a 95% confidence level means that if you were to run the same survey 100 times, you would get similar results 95 of those 100 times. Confidence level is targeted before you start your study because it will affect how big your margin of error is at the end of your study.
Confidence interval	The range of possible values that the population's result would be at the confidence level of the study. This range is the sample result +/- the margin of error.
Statistical significance	The determination of whether your result could be due to random chance or not. The greater the significance, the less due to chance.

Things to remember when determining the size of your sample

When figuring out a sample size, here are things to keep in mind:

- Don’t use a sample size less than 30. It has been statistically proven that 30 is the smallest sample size where an average result of a sample starts to represent the average result of a population.
- The confidence level most commonly used is 95%, but 90% can work in some cases.

Increase the sample size to meet specific needs of your project:

- For a **higher** confidence level, use a larger sample size
- To **decrease** the margin of error, use a larger sample size
- For **greater** statistical significance, use a larger sample size

Note: Sample size calculators use statistical formulas to determine a sample size. More about these are coming up in the course! Stay tuned.

Why a minimum sample of 30?

This recommendation is based on the **Central Limit Theorem (CLT)** in the field of probability and statistics. As sample size increases, the results more closely resemble the normal (bell-shaped) distribution from a large number of samples. A sample of 30 is the smallest sample size for which the CLT is still valid. Researchers who rely on **regression analysis** – statistical methods to determine the relationships between controlled and dependent variables – also prefer a minimum sample of 30.

Sample sizes vary by business problem

Sample size will vary based on the type of business problem you are trying to solve.

For example, if you live in a city with a population of 200,000 and get 180,000 people to respond to a survey, that is a large sample size. But without actually doing that, what would an acceptable, smaller sample size look like?

Would 200 be alright if the people surveyed represented every district in the city?

Answer: It depends on the stakes.

- A sample size of 200 might be large enough if your business problem is to find out how residents felt about the new library
- A sample size of 200 might not be large enough if your business problem is to determine how residents would vote to fund the library

You could probably accept a larger margin of error surveying how residents feel about the new library versus surveying residents about how they would vote to fund it. For that reason, you would most likely use a larger sample size for the voter survey.

Larger sample sizes have a higher cost

You also have to weigh the cost against the benefits of more accurate results with a larger sample size. Someone who is trying to understand consumer preferences for a new line of products wouldn't need as large a sample size as someone who is trying to understand the effects of a new drug. For drug safety, the benefits outweigh the cost of using a larger sample size. But for consumer preferences, a smaller sample size at a lower cost could provide good enough results.

Knowing the basics is helpful

Knowing the basics will help you make the right choices when it comes to sample size. You can always raise concerns if you come across a sample size that is too small. A sample size calculator is also a great tool for this. Sample size calculators let you enter a desired confidence level and margin of error for a given population size. They then calculate the sample size needed to statistically achieve those results.

When data isn't readily available

Earlier, you learned how you can still do an analysis using proxy data if you have no data. You might have some questions about proxy data, so this reading will give you a few more examples of the types of datasets that can serve as alternate data sources.

Proxy data examples

Sometimes the data to support a business objective isn't readily available. This is when proxy data is useful. Take a look at the following scenarios and where proxy data comes in for each example:

Business scenario	How proxy data can be used
A new car model was just launched a few days ago and the auto dealership can't wait until the end of the month for sales data to come in. They want sales projections now.	The analyst proxies the number of clicks to the car specifications on the dealership's website as an estimate of potential sales at the dealership.
A brand new plant-based meat product was only recently stocked in grocery stores and the supplier needs to estimate the demand over the next four years.	The analyst proxies the sales data for a turkey substitute made out of tofu that has been on the market for several years.
The Chamber of Commerce wants to know how a tourism campaign is going to impact travel to their city, but the results from the campaign aren't publicly available yet.	The analyst proxies the historical data for airline bookings to the city one to three months after a similar campaign was run six months earlier.

Open (public) datasets

If you are part of a large organization, you might have access to lots of sources of data. But if you are looking for something specific or a little outside your line of business, you can also make use of open or public datasets.

Here's an example. A nasal version of a vaccine was recently made available. A clinic wants to know what to expect for contraindications, but just started collecting first-party data from its patients. A **contraindication** is a condition that may cause a patient not to take a vaccine due to the harm it would cause them if taken. To estimate the number of possible contraindications, a data analyst proxies an open dataset from a trial of the injection version of the vaccine. The analyst selects a subset of the data with patient profiles most closely matching the makeup of the patients at the clinic.

There are plenty of ways to share and collaborate on data within a community. Kaggle which we previously introduced, has datasets in a variety of formats including the most basic type, Comma Separated Values (CSV) files.

All about margin of error

Margin of error is the maximum amount that the sample results are expected to differ from those of the actual population. More technically, the margin of error defines a range of values below and above the average result for the sample. The average result for the entire population is expected to be within that range. We can better understand margin of error by using some examples below.

Margin of error in baseball

Imagine you are playing baseball and that you are up at bat. The crowd is roaring, and you are getting ready to try to hit the ball. The pitcher delivers a fastball traveling about 90-95mph, which takes about 400 milliseconds (ms) to reach the catcher's glove. You swing and miss the first pitch because your timing was a little off. You wonder if you should have swung slightly earlier or slightly later to hit a home run. That time difference can be considered the margin of error, and it tells us how close or far your timing was from the average home run swing.

Margin of error in marketing

The margin of error is also important in marketing. Let's use A/B testing as an example. **A/B testing** (or split testing) tests two variations of the same web page to determine which page is more successful in attracting user traffic and generating revenue. User traffic that gets monetized is known as the **conversion rate**. A/B testing allows marketers to test emails, ads, and landing pages to find the data behind what is working and what isn't working. Marketers use the **confidence interval** (determined by the conversion rate and the margin of error) to understand the results.

For example, suppose you are conducting an A/B test to compare the effectiveness of two different email subject lines to entice people to open the email. You find that subject line A: "Special offer just for you" resulted in a 5% open rate compared to subject line B: "Don't miss this opportunity" at 3%.

Does that mean subject line A is better than subject line B? It depends on your margin of error. If the margin of error was 2%, then subject line A's actual open rate or confidence interval is somewhere between 3% and 7%. Since the lower end of the interval overlaps with subject line B's results at 3%, you can't conclude that there is a statistically significant difference between subject line A and B. Examining the margin of error is important when making conclusions based on your test results.

Want to calculate your margin of error?

All you need is population size, confidence level, and sample size. In order to better understand this calculator, review these terms:

- **Confidence level**: A percentage indicating how likely your sample accurately reflects the greater population
- **Population**: The total number you pull your sample from
- **Sample**: A part of a population that is representative of the population
- **Margin of error**: The maximum amount that the sample results are expected to differ from those of the actual population

In most cases, a 90% or 95% confidence level is used. But, depending on your industry, you might want to set a stricter confidence level. A 99%

confidence level is reasonable in some industries, such as the pharmaceutical industry.

Glossary terms from module 1

Terms and definitions for Course 4, Module 1

Confidence interval: A range of values that conveys how likely a statistical estimate reflects the population

Confidence level: The probability that a sample size accurately reflects the greater population

Consistency: The degree to which data is repeatable from different points of entry or collection

Cross-field validation: A process that ensures certain conditions for multiple data fields are satisfied

Data constraints: The criteria that determine whether a piece of a data is clean and valid

Data integrity: The accuracy, completeness, consistency, and trustworthiness of data throughout its life cycle

Data manipulation: The process of changing data to make it more organized and easier to read

Data replication: The process of storing data in multiple locations

DATEDIF: A spreadsheet function that calculates the number of days, months, or years between two dates

Estimated response rate: The average number of people who typically complete a survey

Hypothesis testing: A process to determine if a survey or experiment has meaningful results

Mandatory: A data value that cannot be left blank or empty

Margin of error: The maximum amount that the sample results are expected to differ from those of the actual population

Random sampling: A way of selecting a sample from a population so that every possible type of the sample has an equal chance of being chosen

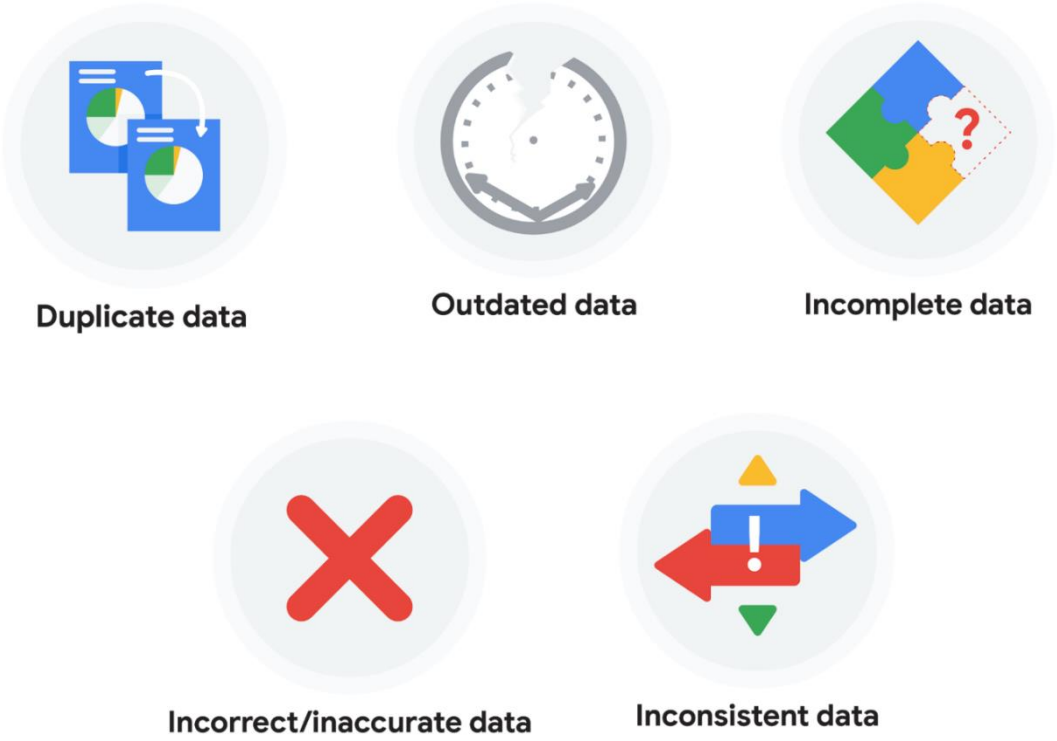
Regular expression (RegEx): A rule that says the values in a table must match a prescribed pattern

Module 2
What is dirty data?

Earlier, we discussed that **dirty data** is data that is incomplete, incorrect, or irrelevant to the problem you are trying to solve. This reading summarizes:

- Types of dirty data you may encounter
- What may have caused the data to become dirty
- How dirty data is harmful to businesses

Types of dirty data



Duplicate data

Description	Possible causes	Potential harm to businesses
Any data record that shows up more than once	Manual data entry, batch data imports, or data migration	Skewed metrics or analyses, inflated or inaccurate counts or predictions, or confusion during data retrieval

Outdated data

Description	Possible causes	Potential harm to businesses
Any data that is old which should be replaced with newer and more accurate information	People changing roles or companies, or software and systems becoming obsolete	Inaccurate insights, decision-making, and analytics

Incomplete data

Description	Possible causes	Potential harm to businesses
Any data that is missing important fields	Improper data collection or incorrect data entry	Decreased productivity, inaccurate insights, or inability to complete essential services

Incorrect/inaccurate data

Description	Possible causes	Potential harm to businesses
Any data that is complete but inaccurate	Human error inserted during data input, fake information, or mock data	Inaccurate insights or decision-making based on bad information resulting in revenue loss

Inconsistent data

Description	Possible causes	Potential harm to businesses
Any data that uses different formats to represent the same thing	Data stored incorrectly or errors inserted during data transfer	Contradictory data points leading to confusion or inability to classify or segment customers

Common data-cleaning pitfalls

In this reading, you will learn the importance of data cleaning and how to identify common mistakes. Some of the errors you might come across while cleaning your data could include:

Common mistakes to avoid

Not checking for spelling errors: Misspellings can be as simple as typing or input errors. Most of the time the wrong spelling or common grammatical errors can be detected, but it gets harder with things like names or addresses. For example, if you are working with a spreadsheet table of

customer data, you might come across a customer named “John” whose name has been input incorrectly as “Jon” in some places. The spreadsheet’s spellcheck probably won’t flag this, so if you don’t double-check for spelling errors and catch this, your analysis will have mistakes in it.

Forgetting to document errors: Documenting your errors can be a big time saver, as it helps you avoid those errors in the future by showing you how you resolved them. For example, you might find an error in a formula in your spreadsheet. You discover that some of the dates in one of your columns haven’t been formatted correctly. If you make a note of this fix, you can reference it the next time your formula is broken, and get a head start on troubleshooting. Documenting your errors also helps you keep track of changes in your work, so that you can backtrack if a fix didn’t work.

Not checking for misfielded values: A misfielded value happens when the values are entered into the wrong field. These values might still be formatted correctly, which makes them harder to catch if you aren’t careful. For example, you might have a dataset with columns for cities and countries. These are the same type of data, so they are easy to mix up. But if you were trying to find all of the instances of Spain in the country column, and Spain had mistakenly been entered into the city column, you would miss key data points. Making sure your data has been entered correctly is key to accurate, complete analysis.

Overlooking missing values: Missing values in your dataset can create errors and give you inaccurate conclusions. For example, if you were trying to get the total number of sales from the last three months, but a week of transactions were missing, your calculations would be inaccurate. As a best practice, try to keep your data as clean as possible by maintaining completeness and consistency.

Only looking at a subset of the data: It is important to think about all of the relevant data when you are cleaning. This helps make sure you understand the whole story the data is telling, and that you are paying attention to all possible errors. For example, if you are working with data about bird migration patterns from different sources, but you only clean one source, you might not realize that some of the data is being repeated. This will cause problems in your analysis later on. If you want to avoid common errors like duplicates, each field of your data requires equal attention.

Losing track of business objectives: When you are cleaning data, you might make new and interesting discoveries about your dataset-- but you don't want those discoveries to distract you from the task at hand. For example, if you were working with weather data to find the average number of rainy days in your city, you might notice some interesting patterns about snowfall, too. That is really interesting, but it isn't related to the question you are trying to answer right now. Being curious is great! But try not to let it distract you from the task at hand.

Not fixing the source of the error: Fixing the error itself is important. But if that error is actually part of a bigger problem, you need to find the source of the issue. Otherwise, you will have to keep fixing that same error over and over again. For example, imagine you have a team spreadsheet that tracks everyone's progress. The table keeps breaking because different people are entering different values. You can keep fixing all of these problems one by one, or you can set up your table to streamline data entry so everyone is on the same page. Addressing the source of the errors in your data will save you a lot of time in the long run.

Not analyzing the system prior to data cleaning: If we want to clean our data and avoid future errors, we need to understand the root cause of your dirty data. Imagine you are an auto mechanic. You would find the cause of the problem before you started fixing the car, right? The same goes for data. First, you figure out where the errors come from. Maybe it is from a data entry error, not setting up a spell check, lack of formats, or from duplicates. Then, once you understand where bad data comes from, you can control it and keep your data clean.

Not backing up your data prior to data cleaning: It is always good to be proactive and create your data backup before you start your data clean-up. If your program crashes, or if your changes cause a problem in your dataset, you can always go back to the saved version and restore it. The simple procedure of backing up your data can save you hours of work-- and most importantly, a headache.

Not accounting for data cleaning in your deadlines/process: All good things take time, and that includes data cleaning. It is important to keep that in mind when going through your process and looking at your deadlines. When you set aside time for data cleaning, it helps you get a more accurate estimate

for ETAs for stakeholders, and can help you know when to request an adjusted ETA.

Develop your approach to cleaning data

As you continue on your data journey, you're likely discovering that data is often messy—and you can expect raw, primary data to be imperfect. In this reading, you'll consider how to develop your personal approach to cleaning data. You will explore the idea of a cleaning checklist, which you can use to guide your cleaning process. Then, you'll define your preferred methods for cleaning data. By the time you complete this reading, you'll have a better understanding of how to methodically approach the data cleaning process. This will save you time when cleaning data and help you ensure that your data is clean and usable.

Consider your approach to cleaning data

Data cleaning usually requires a lot of time, energy, and attention. But there are two steps you can take before you begin to help streamline your process: creating a cleaning checklist and deciding on your preferred methods. This will help ensure that you know exactly how you want to approach data cleaning and what you need to do to be confident in the integrity of your data.

Your cleaning checklist

Start developing your personal approach to cleaning data by creating a checklist to help you identify problems in your data efficiently and identify the scale and scope of your dataset. Think of this checklist as your default "what to search for" list.

Here are some examples of common data cleaning tasks you could include in your checklist:

- **Determine the size of the dataset:** Large datasets may have more data quality issues and take longer to process. This may impact your choice of data cleaning techniques and how much time to allocate to the project.

- **Determine the number of categories or labels:** By understanding the number and nature of categories and labels in a dataset, you can better understand the diversity of the dataset. This understanding also helps inform data merging and migration strategies.

- **Identify missing data:** Recognizing missing data helps you understand data quality so you can take appropriate steps to remediate the problem. Data integrity is important for accurate and unbiased analysis.

- **Identify unformatted data:** Identifying improperly or inconsistently formatted data helps analysts ensure data uniformity. This is essential for accurate analysis and visualization.

- **Explore the different data types:** Understanding the types of data in your dataset (for instance, numerical, categorical, text) helps you select appropriate cleaning methods and apply relevant data analysis techniques.

  There might be other data cleaning tasks you've been learning about that you also want to prioritize in your checklist. Your checklist is an opportunity for you to define exactly what you want to remember about cleaning your data; feel free to make it your own.

Your preferred cleaning methods

In addition to creating a checklist, identify which actions or tools you prefer using when cleaning data. You'll use these tools and techniques with each new dataset—or whenever you encounter issues in a dataset—so this list should be compatible with your checklist.

For example, suppose you have a large dataset with missing data. You'll want to know how to check for missing data in larger datasets, and how you plan to handle any missing data, before you start cleaning. Outlining your preferred methods can save you lots of time and energy.

Glossary terms from module 2

Terms and definitions for Course 4, Module 2

Compatibility: How well two or more datasets are able to work together

CONCATENATE: A spreadsheet function that joins together two or more text strings

Conditional formatting: A spreadsheet tool that changes how cells appear when values meet specific conditions

Data engineer: A professional who transforms data into a useful format for analysis and gives it a reliable infrastructure

Data mapping: The process of matching fields from one data source to another

Data merging: The process of combining two or more datasets into a single dataset

Data validation: A tool for checking the accuracy and quality of data

Data warehousing specialist: A professional who develops processes and procedures to effectively store and organize data

Delimiter: A character that indicates the beginning or end of a data item

Field length: A tool for determining how many characters can be keyed into a spreadsheet field

Incomplete data: Data that is missing important fields

Inconsistent data: Data that uses different formats to represent the same thing

LEFT: A function that returns a set number of characters from the left side of a text string

LEN: A function that returns the length of a text string by counting the number of

Merger: An agreement that unites two organizations into a single new one

MID: A function that returns a segment from the middle of a text string

Split: A function that divides text around a specified character and puts each fragment into a new, separate cell

TRIM: A function that removes leading, trailing, and repeated spaces in data

Module 3
How a junior data analyst uses SQL

In this reading, you will learn more about how to decide when to use SQL, or Structured Query Language. As a data analyst, you will be tasked with handling a lot of data, and SQL is one of the tools that can help make your work a lot easier. SQL is the primary way data analysts extract data from databases. As a data analyst, you will work with databases all the time, which is why SQL is such a key skill. Let's follow along as a junior data analyst uses SQL to solve a business task.

The business task and context

The junior data analyst in this example works for a social media company. A new business model was implemented on February 15, 2020 and the company wants to understand how their user-growth compares to the previous year. Specifically, the data analyst was asked to find out how many users have joined since February 15, 2020.

Spreadsheets functions and formulas or SQL queries?

Before they can address this question, this data analyst needs to choose what tool to use. First, they have to think about where the data lives. If it is stored in a database, then SQL is the best tool for the job. But if it is stored in a spreadsheet, then they will have to perform their analysis in that spreadsheet. In that scenario, they could create a pivot table of the data and then apply specific formulas and filters to their data until they were given the number of users that joined after February 15th. It isn't a really complicated process, but it would involve a lot of steps.

Glossary terms from module 3

Terms and definitions for Course 4, Module 3

CAST: A SQL function that converts data from one datatype to another

COALESCE: A SQL function that returns non-null values in a list

CONCAT: A SQL function that adds strings together to create new text strings that can be used as unique keys

DISTINCT: A keyword that is added to a SQL SELECT statement to retrieve only non-duplicate entries

Float: A number that contains a decimal

Substring: A subset of a text string

Typecasting: Converting data from one type to another

Module 4

Data-cleaning verification checklist

This reading will give you a checklist of common problems you can refer to when doing your data cleaning verification, no matter what tool you are using. When it comes to data cleaning verification, there is no one-size-fits-all approach or a single checklist that can be universally applied to all projects. Each project has its own organization and data requirements that lead to a unique list of things to run through for verification.

Keep in mind, as you receive more data or a better understanding of the project goal(s), you might want to revisit some or all of these steps.

Correct the most common problems

Make sure you identified the most common problems and corrected them, including:

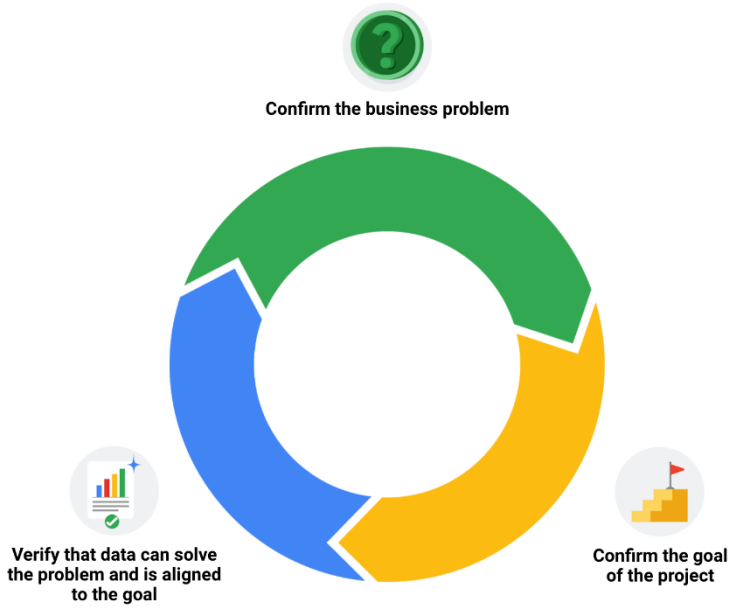
- **Sources of errors**: Did you use the right tools and functions to find the source of the errors in your dataset?
- **Null data**: Did you search for NULLs using conditional formatting and filters?
- **Misspelled words**: Did you locate all misspellings?
- **Mistyped numbers**: Did you double-check that your numeric data has been entered correctly?
- **Extra spaces and characters**: Did you remove any extra spaces or characters using the **TRIM** function?

- **Duplicates**: Did you remove duplicates in spreadsheets using the **Remove Duplicates** function or **DISTINCT** in SQL?
- **Mismatched data types**: Did you check that numeric, date, and string data are typecast correctly?
- **Messy (inconsistent) strings**: Did you make sure that all of your strings are consistent and meaningful?
- **Messy (inconsistent) date formats**: Did you format the dates consistently throughout your dataset?
- **Misleading variable labels (columns)**: Did you name your columns meaningfully?
- **Truncated data:** Did you check for truncated or missing data that needs correction?
- **Business Logic**: Did you check that the data makes sense given your knowledge of the business?

Review the goal of your project

Once you have finished these data cleaning tasks, it is a good idea to review the goal of your project and confirm that your data is still aligned with that goal. This is a continuous process that you will do throughout your project-- but here are three steps you can keep in mind while thinking about this:

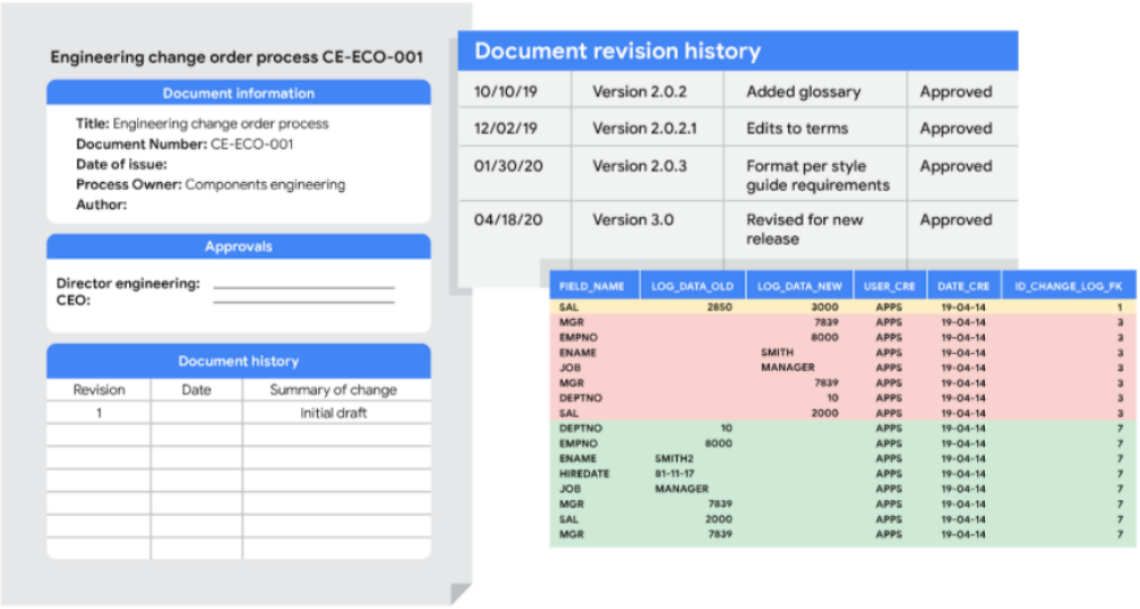
- Confirm the business problem
- Confirm the goal of the project
- Verify that data can solve the problem and is aligned to the goal



Embrace Changelogs

What do engineers, writers, and data analysts have in common? Change.

Engineers use **engineering change orders** (ECOs) to keep track of new product design details and proposed changes to existing products. Writers use **document revision histories** to keep track of changes to document flow and edits. And data analysts use **changelogs** to keep track of data transformation and cleaning. Here are some examples of these:



Engineering change order process CE-ECO-001

Document information

Title: Engineering change order process
Document Number: CE-ECO-001
Date of issue:
Process Owner: Components engineering
Author:

Approvals

Director engineering: _____
CEO: _____

Document history

Revision	Date	Summary of change
1		Initial draft

Document revision history

10/10/19	Version 2.0.2	Added glossary	Approved
12/02/19	Version 2.0.2.1	Edits to terms	Approved
01/30/20	Version 2.0.3	Format per style guide requirements	Approved
04/18/20	Version 3.0	Revised for new release	Approved

FIELD_NAME	LOG_DATA_OLD	LOG_DATA_NEW	USER_CRE	DATE_CRE	ID_CHANGE_LOG_FK
SAL	2850	3000	APPS	19-04-14	1
MGR		7839	APPS	19-04-14	3
EMPNO		8000	APPS	19-04-14	3
ENAME		SMITH	APPS	19-04-14	3
JOB		MANAGER	APPS	19-04-14	3
MGR		7839	APPS	19-04-14	3
DEPTNO		10	APPS	19-04-14	3
SAL		2000	APPS	19-04-14	3
DEPTNO	10		APPS	19-04-14	7
EMPNO	8000		APPS	19-04-14	7
ENAME	SMITH2		APPS	19-04-14	7
HIREDATE	81-11-17		APPS	19-04-14	7
JOB	MANAGER		APPS	19-04-14	7
MGR	7839		APPS	19-04-14	7
SAL	2000		APPS	19-04-14	7
MGR	7839		APPS	19-04-14	7

Automated version control takes you most of the way

Most software applications have a kind of history tracking built in. For example, in Google sheets, you can check the version history of an entire sheet or an individual cell and go back to an earlier version. In Microsoft Excel, you can use a feature called **Track Changes**. And in BigQuery, you can view the history to check what has changed.

Here's how it works:

Google Sheets	1. Right-click the cell and select **Show edit history**. 2. Click the left-arrow < or right arrow > to move backward and forward in the history as needed.
Microsoft Excel	1. If Track Changes has been enabled for the spreadsheet: click **Review**. 2. Under **Track Changes**, click the **Accept/Reject Changes** option to accept or reject any change made.
BigQuery	Bring up a previous version (without reverting to it) and figure out what changed by comparing it to the current version.

Changelogs take you down the last mile

A **changelog** can build on your automated version history by giving you an even more detailed record of your work. This is where data analysts record all the changes they make to the data. Here is another way of looking at it. Version histories record *what* was done in a data change for a project, but don't tell us *why*. Changelogs are super useful for helping us understand the reasons changes have been made. Changelogs have no set format and you can even make your entries in a blank document. But if you are using a shared changelog, it is best to agree with other data analysts on the format of all your log entries.

Typically, a changelog records:

- Data, file, formula, query, or any other component that changed
- Description of what changed
- Date of the change
- Person who made the change
- Person who approved the change
- Version number
- Reason for the change

Let’s say you made a change to a formula in a spreadsheet because you observed it in another report and you wanted your data to match and be consistent. If you found out later that the report was actually using the wrong formula, an automated version history would help you *undo* the change. But if you also recorded the reason for the change in a changelog, you could go back to the creators of the report and let them know about the incorrect formula. If the change happened a while ago, you might not remember who to follow up with. Fortunately, your changelog would have that information ready for you! By following up, you would ensure data integrity outside your project. You would also be showing personal integrity as someone who can be trusted with data. That is the power of a changelog!

Finally, a changelog is important for when lots of changes to a spreadsheet or query have been made. Imagine an analyst made four changes and the change they want to revert to is change #2. Instead of clicking the undo feature three times to undo change #2 (and losing changes #3 and #4), the analyst can undo just change #2 and keep all the other changes. Now, our example was for just 4 changes, but try to think about how important that changelog would be if there were hundreds of changes to keep track of.

Bonus tip

If an analyst is making changes to an existing SQL query that is shared across the company, the company most likely uses what is called a **version control system**. An example might be a query that pulls daily revenue to build a dashboard for senior management.

Here's how a version control system affects a change to a query:

1. A company has official versions of important queries in their **version control system**.
2. An analyst makes sure the most up-to-date version of the query is the one they will change. This is called **syncing**
3. The analyst makes a change to the query.
4. The analyst might ask someone to review this change. This is called a **code review** and can be informally or formally done. An informal review could be as simple as asking a senior analyst to take a look at the change.
5. After a reviewer approves the change, the analyst submits the updated version of the query to a repository in the company's version control system. This is called a **code commit**. A best practice is to document exactly what the change was and why it was made in a comments area. Going back to our example of a query that pulls daily revenue, a comment might be: *Updated revenue to include revenue coming from the new product, Calypso*.
6. After the change is **submitted**, everyone else in the company will be able to access and use this new query when they **sync** to the most up-to-date queries stored in the version control system.
7. If the query has a problem or business needs change, the analyst can ***undo*** the change to the query using the version control system. The analyst can look at a chronological list of all changes made to the query and who made each change. Then, after finding their own change, the analyst can **revert** to the previous version.
8. The query is back to what it was before the analyst made the change. And everyone at the company sees this reverted, original query, too.

```
# Changelog
This file contains the notable changes to the project

Version 1.0.0 (02-23-2019)
## New
    - Added column classifiers (Date, Time, PerUnitCost, TotalCost, et
c. )
    - Added Column "AveCost" to track average item cost

## Changes
    - Changed date format to MM-DD-YYYY
    - Removal of whitespace (cosmetic)

## Fixes
    - Fixed misalignment in Column "TotalCost" where some rows did not
 match with correct dates
    - Fixed SUM to run over entire column instead of partial
```

Glossary terms from module 4

Terms and definitions for Course 4, Module 4

CASE: A SQL statement that returns records that meet conditions by including an if/then statement in a query

Changelog: A file containing a chronologically ordered list of modifications made to a project

COUNTA: A spreadsheet function that counts the total number of values within a specified range

Find and replace: A tool that finds a specified search term and replaces it with something else

Verification: A process to confirm that a data-cleaning effort was well executed and the resulting data is accurate and reliable