Week 1: Ensuring data integrity

Focus on integrity

Data integrity: the accuracy, completeness, consistency, and trustworthiness of data throughout its lifecycle.

Ways to ensure Data integrity:

- **Data replication:** Two versions of a dataset can introduce inconsistent results. A final audit of results would be essential to reveal what happened and correct all dates.
- **Data transfer:** Due to error which can exists in the transferring process, the data needs to be cleaned
- **Data manipulation**: To make data more organized and easier to read

Data constraints

Data constraint	Definition	Examples
Data type	Values must be of a certain type: date, number, percentage, Boolean, etc.	If the data type is a date, a single number like 30 would fail the constraint and be invalid
Data range	Values must fall between predefined maximum and minimum values	If the data range is 10-20, a value of 30 would fail the constraint and be invalid
Mandatory	Values can't be left blank or empty	If age is mandatory, that value must be filled in
Unique	Values can't have a duplicate	Two people can't have the same mobile phone number within the same service area
Regular expression (regex) patterns	Values must match a prescribed pattern	A phone number must match ###-###-#### (no other characters allowed)
Cross-field validation	Certain conditions for multiple fields must be satisfied	Values are percentages and values from multiple fields must add up to 100%
Primary-key	(Databases only) value must be unique per column	A database table can't have two rows with the same primary key value. A primary key is an identifier in a database that references a column in which each

		value is unique. More information about primary and foreign keys is provided later in the program.
Set-membership	(Databases only) values for a column must come from a set of discrete values	Value for a column must be set to Yes, No, or Not Applicable
Foreign-key	(Databases only) values for a column must be unique values coming from a column in another table	In a U.S. taxpayer database, the State column must be a valid state or territory with the set of acceptable values defined in a separate States table
Accuracy	The degree to which the data conforms to the actual entity being measured or described	If values for zip codes are validated by street location, the accuracy of the data goes up.
Completeness	The degree to which the data contains all desired components or measures	If data for personal profiles required hair and eye color, and both are collected, the data is complete.
Consistency	The degree to which the data is repeatable from different points of entry or collection	If a customer has the same address in the sales and repair databases, the data is consistent.

Balancing objectives with data integrity

We need to align data integrity with business objectives.

- When there is clean data and good alignment, you can get accurate insights and make conclusions the data supports.
- If there is good alignment but the data needs to be cleaned, **clean the data** before you perform your analysis.
- If the data only partially aligns with an objective, think about how you could **modify** the objective, or **use data constraints** to make sure that the subset of data better aligns with the business objective. (eg: when process a variable but found the data divided in two scenarios)

Dealing with insufficient data

Types of insufficient data

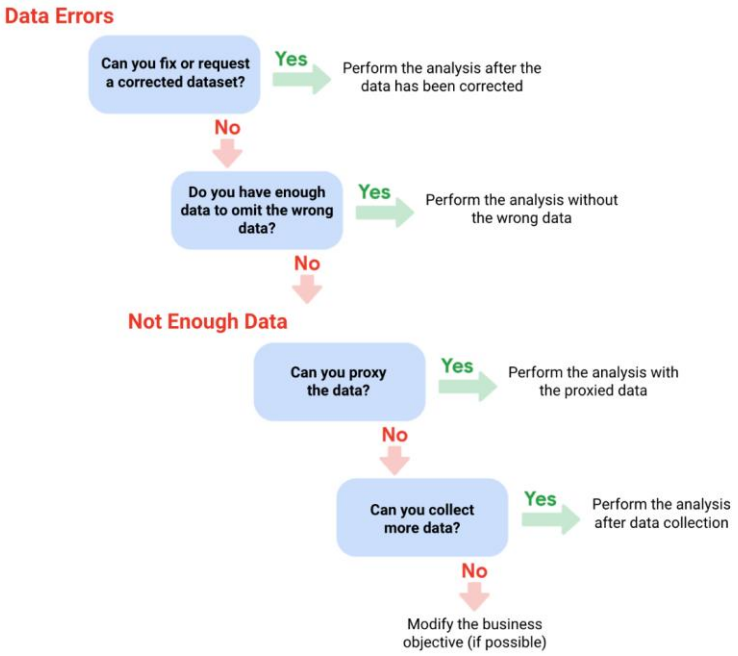
- Data from just one source

- Data that keeps updating: the data is still incoming and might not be complete. Eg: brand new tourist attraction => wait of adj the obj
- Outdated data: your data could be older and no longer be relevant => best: find a new data set to work with
- Geographically limited

Ways to address:

- Identify trends with the available data
- Wait for more data if time allows
- Talk with stakeholders and adjust your objective
- Look for a new data set.

Other issues:



To sum up: You should complete the following tasks before analyzing data:

1. Determine data integrity by assessing the overall accuracy, consistency, and completeness of the data.

2. Connect objectives to data by understanding how your business objectives can be served by an investigation into the data.

3. Know when to stop collecting data.

Testing your data

"Statistical power can be calculated and reported for a completed experiment to comment on the confidence one might have in the conclusions drawn from the results of the study. It can also be used as a tool to estimate the number of observations or sample size required in order to detect an effect in an experiment."

Make sure statistical power is >= 80%. In order to do that, we need to consider all factor can prevent statistical power to make sure its align with the business objective

Commented [NL1]: Which means 80% correct result

Sample

Choose a sample size >= 30 (Central Limit Theorem)

Sample sizes vary by business problem.

Commented [NL2]: •Eg: 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

Sample Size Calculator: the calculated sample size is the **minimum** number to achieve what you input for confidence level and margin of error.

If you are working with a survey, you will also need to think about the estimated response rate to figure out how many surveys you will need to send out. For example, if you need a sample size of 100 individuals and your estimated response rate is 10%, you will need to send your survey to 1,000 individuals to get the 100 responses you need for your analysis.

Most of industries hope 90% or 95% confidence level.

Margin of Error Calculator

Margin of error: maximum amount that the sample results are expected to differ from those of the actual population. For example, a 5% margin of error implies that 55 to 65% people agree (when the result is 60%)

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.

Week 2: Understanding clean data

Poor quality data: most comes from human error (incorrect data, repeated data, blank data, inconsistent data format, etc.)

Remember: no data set is perfect, even its prepared by DE, or data warehouse specialist -> always clean data 1st

Clean data: data that's **complete, correct,** and **relevant** to the problem you're trying to solve.

Dirty data

Dirty data: data that is incomplete, incorrect, or irrelevant to the problem you are trying to solve.

Types of dirty data

	Description	Possible causes
Duplicate data	Any data record that shows up more than once	Manual data entry, batch data imports, or data migration
Outdated data	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
Incomplete data	Any data that is missing important fields (Null: empty field)	Improper data collection or incorrect data entry
Incorrect/inaccurate data	Any data that is complete but inaccurate	Human error inserted during data input, fake information, or mock data
Inconsistent data	Any data that uses different formats to represent the same thing	Data stored incorrectly or errors inserted during data transfer

Recognize and remedy dirty data

Data validation: check the data (integrity principles) and set standards (defined business rules or constraints) before importing data to avoid error

Some common tools and techniques:

Note: Before removing unwanted data, it's always a good practice to make a copy of the data set. That way, if you remove something that you end up needing in the future, you can easily access it and put it back in the data set.

Format: (in spreadsheet) clear formats

Cleaning data from multiple sources: combine 2 data sets => ensure consistence

➔ Ask:

- Do I have all the data I need?
- Does the data I need exist within these datasets?
- Does the data need to be cleaned, or are they ready for me to use?
- Are the datasets cleaned to the same standard?

Common data-cleaning pitfalls

- **Not checking for spelling 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 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.
- **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.
- **Only looking at a subset of the data**: 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.
- **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.
- **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. 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**
- **Not accounting for data cleaning in your deadlines/process**: It is important to keep that in mind when going through your process and looking at your deadlines.

Additional resources

Refer to these "top ten" lists for data cleaning in Microsoft Excel and Google Sheets to help you avoid the most common mistakes:

- Top ten ways to clean your data: Review an orderly guide to data cleaning in Microsoft Excel.
- 10 Google Workspace tips to clean up data: Learn best practices for data cleaning in Google Sheets.

Before processing data, we need to determine what to do what the blank row, trim the data, format and delete format, make sure everything is on its right format

Name of the formula	Purpose
Countif	Check to see if there are any values that inaccurate (out of our expectation)
LEN	if you have a certain piece of information in your spreadsheet that you know must contain a certain length.
LEFT; RIGHT, MID	If you only want to work with a certain character in a text string
CONCATENATE	Joins >=2 text string
TRIM	TRIM is a function that removes leading, trailing, and repeated spaces in data.

Workflow automation

- TechnologyAdvice's **10 of the Best Options for Workflow Automation Software**

Data cleaning checklist	Preferred cleaning methods
Missing data	Reach out to fill, or delete data
Unformatted data	TRIM
The different data types	Format again, delete format
Checking data (is there are any range/conditions that a column has?, etc)	LEN, COUNTIF, CONCATENATE, etc,

---

Week 3: Cleaning data using SQL

Data that is already in a spreadsheet => use spreadsheets

Data stored in a database => SQL

Features of Spreadsheets	**Features of SQL Databases**
Smaller data sets	Larger datasets
Enter data manually	Access tables across a database
Create graphs and visualizations in the same program	Prepare data for further analysis in another software
Built-in spell check and other useful functions	Fast and powerful functionality
Best when working solo on a project	Great for collaborative work and tracking queries run by all users

Some basic command

```sql
UPDATE customer_data.customer_address
SET address = '123 New Address'
WHERE customer_id = 2645
```

To update/ change data

```sql
INSERT INTO customer_data.customer_address
  (customer_id, address, city, state, zipcode, country)
VALUES
  (2645, '333 SQL Road', 'Jackson', 'MI', 49202, 'US')
```

To add more data

If we want to create a new table for this database, we can use the **CREATE TABLE** IF NOT EXISTS statement. (without really create one). To save it, we'll need to download it as a spreadsheet or save the result into a new table.

If you're creating lots of tables within a database, you'll want to use the **DROP TABLE** IF EXISTS statement to clean up the tables you've personally made after finish.

DISTINCT and **COUNT + WHERE:** return the distinct value only (in SELECT)

SUBSTR(value_name, the number starts, total numbers) (Eg: SUBSTR(country,1,2) = 'US')

TRIM(column_names) -> (if an column has a blank) return all columns with that name without blank

CAST(): convert one data type -> another (useful since we don't need to update (manipulate) original data

```sql
SELECT
  CAST(purchase_price AS FLOAT64)
FROM
  customer_data.customer_purchase
ORDER BY
  CAST(purchase_price AS FLOAT64) DESC
```

```sql
SELECT
  CAST(date AS date) as date_only,
  purchase_price
FROM
  customer_data.customer_purchase
WHERE
  date BETWEEN '2020-12-01' AND '2020-12-31'
```

Float: Number contain a decimal

String: text

CONCAST: combine 2 text strings

```sql
SELECT
  CONCAT(product_code, product_color) AS new_product_code
FROM
  customer_data.customer_purchase
WHERE
  product = 'couch'
```

COALESCE: check a column first, then another 2nd (since 1st column might contain null -> Return non-null values in a list)

```
SELECT
  COALESCE(product, product_code) AS product_info
FROM
  customer_data.customer_purchase
```

Week 4: Verifying and reporting cleaning results

Verification is a process to confirm that a data cleaning effort was well- executed and the resulting data is accurate and reliable (double check)

- rechecking your clean dataset
- doing some manual clean ups if needed,
- taking a moment to sit back and really think about the original purpose of the project.
    - Consider the business problem
    - Consider the goal
      ⇨ Do the data help address the problem and achieve the goal
    - Consider the data itself: do the number make sense? (use find and replace; pivot table; CASE statement (SQL):

```
SELECT
  customer_id,
  CASE
    WHEN first_name = 'Tnoy' THEN 'Tony'
    WHEN first_name = 'Tmo' THEN 'Tom'
    WHEN first_name = 'Rachle' THEN 'Rachel'
    ELSE first_name
    END AS cleaned_name
FROM
  customer_data.customer_name
```

Checklist of common error:

- **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?

Advanced functions for speedy data cleaning

Function	Syntax (Google Sheets)	Menu Options (Microsoft Excel)	Primary Use
IMPORTRANGE	=IMPORTRANGE(spreadsheet_url , range_string)	Paste Link (copy the data first)	Imports (pastes) data from one sheet to another and keeps it automatically updated.
QUERY	=QUERY(Sheet and Range, "Select *")	Data > From Other Sources > From Microsoft Query	Enables pseudo SQL (SQL-like) statements or a wizard to import the data.
FILTER	=FILTER(range, condition1, [condition2, ...])	Filter (conditions per column)	Displays only the data that meets the specified conditions.

With query, you can add specific criteria after the SELECT statement by including a WHERE statement. (but remember to make it in “”). QUERY can be used with other functions like SUM and COUNT, but FILTER cant.

Capturing cleaning changes

Documentation: the process of tracking changes, additions, deletions and errors involved in your data cleaning effort.

- ⇨ Use change log: to not only show ‘What we did’, but also ; ‘why we did that’

‘What we did’

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.

‘Why we did that’

Typically, a changelog records this type of information:

- 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

Example of a change log: write text following basic formatting syntax

```
Changelog
is file contains the notable changes to the project

rsion 1.0.0 (02-23-2019)
 New
  - Added column classifiers (Date, Time, PerUnitCost, TotalC
  - 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
  - Fixed SUM to run over entire column instead of partial

```

Version control system

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.

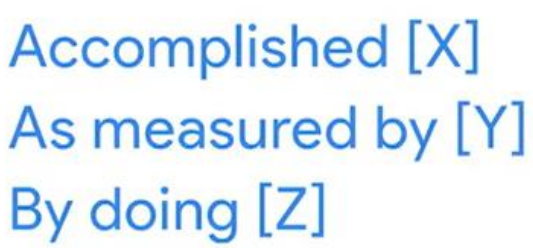
1. A company has official versions of important queries in their **version control system**.
2. An analyst makes sure the most up-to-date version of the query is the one they will change. This is called **syncing**
3. The analyst makes a change to the query.
4. The analyst might ask someone to review 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.

Week 5: Adding data to your resume

When they give you an offer, they want you as much as you want the position. If you have many offer or researching the mean salary, vacation, benefits => there's room for negotiation

A summary can be helpful if you have experience that is not traditional for a data analyst or if you're making a career transition. If you decide to include a summary, keep it to one or two sentences that highlight your strengths and how you can help the company you're applying to. You'll also want to make sure your summary includes positive words about yourself, like dedicated and proactive. You can support those words with data, like the number of years you've worked or the tools you're experienced in like SQL and spreadsheets.

Mix-match your resume and requirements in the JD. Formula:



eg: Selected as one of 275 participants nationwide for this 12-month professional development program for high-achieving talent based on leadership potential and academic success.

earned little-known website over 2,000 new clicks through strategic blogging."

Make your resume unique for DA

Summary:

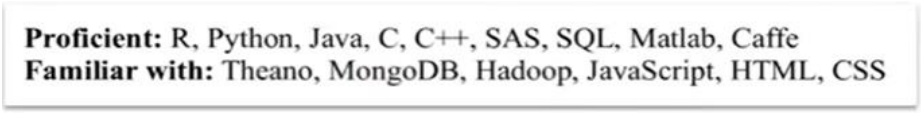
- Problem
- Action
- Result

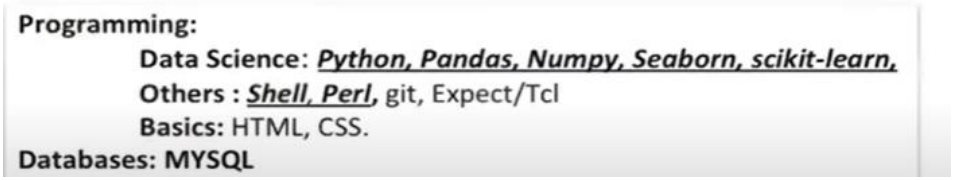
Don't include Objective

Skills – Do list technical skill that job mention (2-4 bullet points)

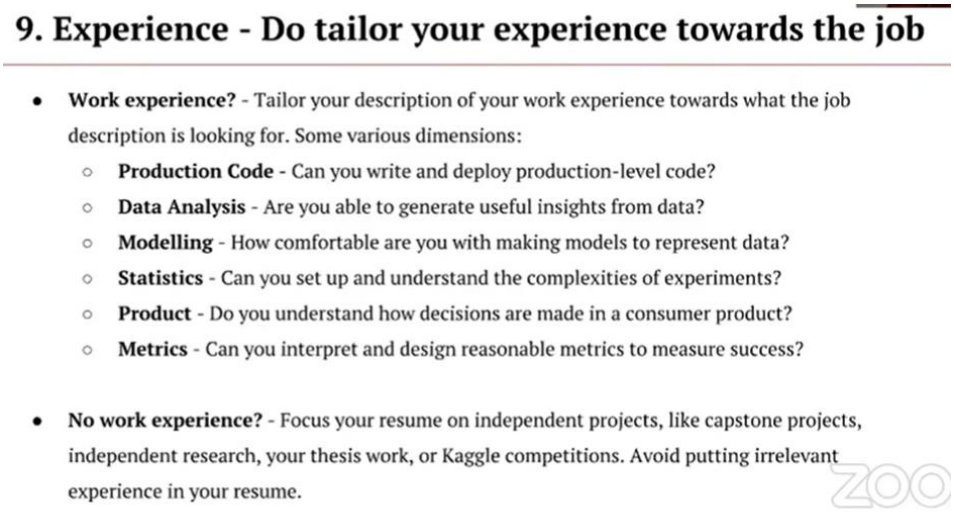
- Strong analytical skills
- Pattern recognition
- Relational databases and SQL
- Strong data visualization skills

- Proficiency with spreadsheets, SQL, R (or Python), and Tableau





Projects – Don't list common project or homework. Show your results and include links (competition, links to your projects,..)



Translating past work experiences

Showing you have

- communication skills: highlight how your effective communication skills have helped you, specific presentations you've made and the outcomes of those presentations, and you can even include the audience for your presentations, especially if you present it to large groups or people in senior positions. After listing job details, like the place and length of employment, you might add something like, "effectively implemented and communicated daily workflow to fellow team members, resulting in an 15% increase in productivity."

- Problem solving skills: PAR
- Teamwork: add what you did for team, and the whole company
- Adaptability
- Attention to details