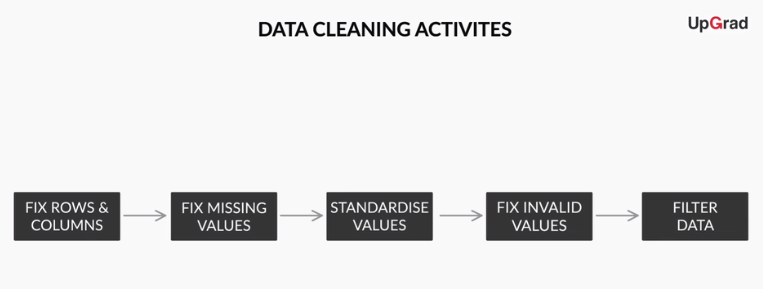
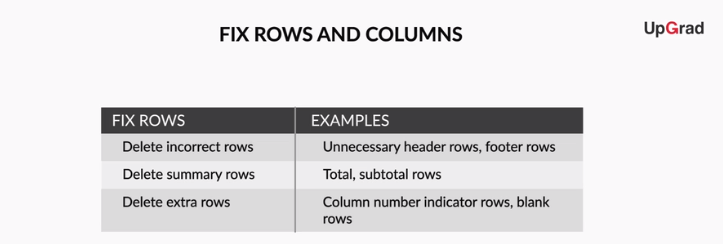
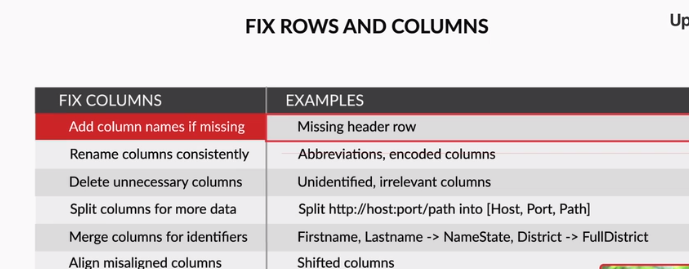
Data Cleaning:

Fixing Rows and Columns







**Checklist for Fixing Rows**

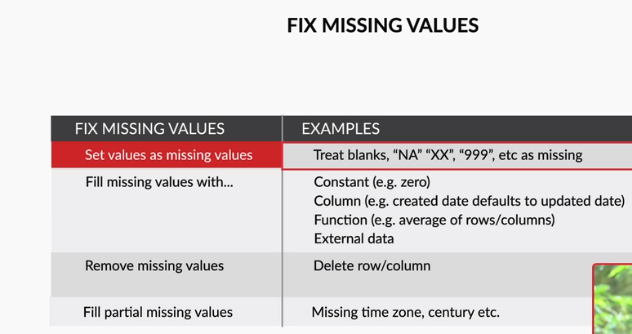
* Delete summary rows: Total, Subtotal rows
* Delete incorrect rows: Header rows, Footer rows
* Delete extra rows: Column number, indicators, Blank rows, Page No.

**Checklist for Fixing Columns**

* Merge columns for creating unique identifiers if needed: E.g. Merge State, City into Full address
* Split columns for more data: Split address to get State and City to analyse each separately
* Add column names: Add column names if missing
* Rename columns consistently: Abbreviations, encoded columns
* Delete columns: Delete unnecessary columns
* Align misaligned columns: Dataset may have shifted columns

Save this checklist for future reference:

Missing Values:



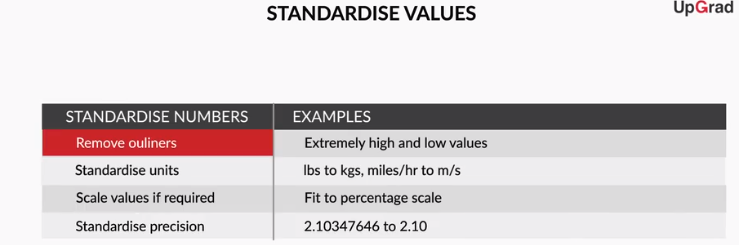
Values may be missing for multiple reasons such as non-response due to sensitivity of information, data entry error, censoring, etc. Without understanding the reason, replacing missing values might lead to a faulty analysis.

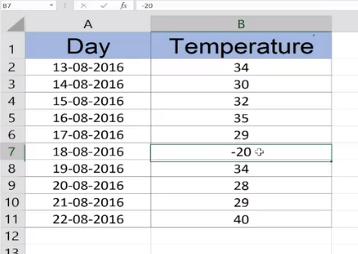
In case you can add information from reliable external sources, you should use it to replace missing values. But often, it is better to let missing values be and continue with the analysis rather than extrapolate the available information.

Let us summarise how to deal with missing values:

* Set values as missing values: Identify values that indicate missing data, and yet are not recognised by the software as such, e.g treat blank strings, "NA", "XX", "999", etc. as missing.
* Adding is good, exaggerating is bad: You should try to get information from reliable external sources as much as possible, but if you can’t, then it is better to keep missing values as such rather than exaggerating the existing rows/columns.
* Delete rows, columns: Rows could be deleted if the number of missing values are insignificant in number, as this would not impact the analysis. Columns could be removed if the missing values are quite significant in number.
* Fill partial missing values using business judgement: Missing time zone, century, etc. These values are easily identifiable.

**Standardizing Values**

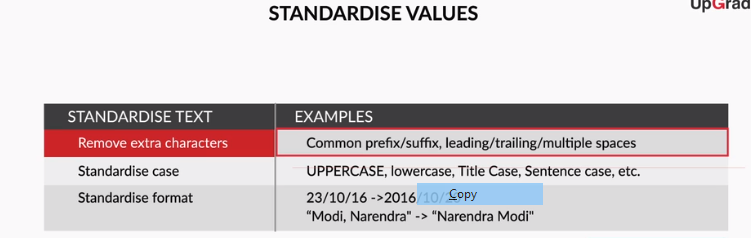




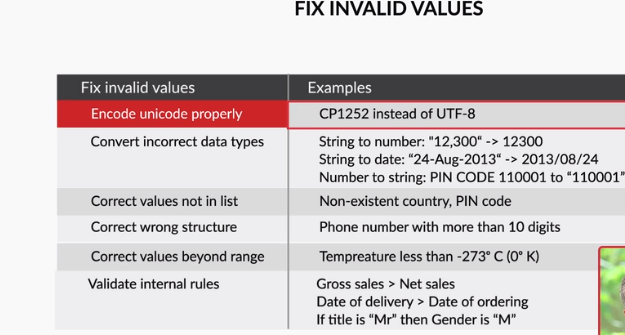
Outlier is -20

Let’s summarize what you learnt about standardizing variables. You could use this as a checklist for future data cleaning exercises.

* **Standardize units**: Ensure all observations under a variable have a common and consistent unit, e.g. convert lbs to kgs, miles/hr to km/hr, etc.
* **Scale values if required**:  Make sure the observations under a variable have a common scale
* **Standardize precision** for better presentation of data, e.g. 4.5312341 kgs to 4.53 kgs.
* **Remove outliers**: Remove high and low values that would disproportionately affect the results of your analysis.



**Invalid Values**

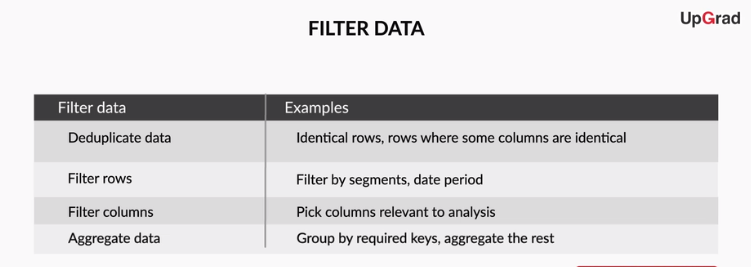


If you have an invalid value problem, and you do not know what accurate values could replace the invalid values, it is recommended to treat these values as missing. E.g. in the case of a string “tr8ml” in a Contact column, it is recommended to remove the invalid value and treat it as a missing value.

* **Encode unicode properly**: In case the data is being read as junk characters, try to change encoding, E.g. CP1252 instead of UTF-8.
* **Convert incorrect data types**: Correct the incorrect data types to the correct data types for ease of analysis. E.g. if numeric values are stored as strings, it would not be possible to calculate metrics such as mean, median, etc. Some of the common data type corrections are — string to number: "12,300" to “12300”; string to date: "2013-Aug" to “2013/08”; number to string: “PIN Code 110001” to "110001"; etc.
* **Correct values that go beyond**range: If some of the values are beyond logical range, e.g. temperature less than -273° C (0° K), you would need to correct them as required. A close look would help you check if there is scope for correction, or if the value needs to be removed.
* **Correct values not in the list**: Remove values that don’t belong to a list. E.g. In a data set containing blood groups of individuals, strings “E” or “F” are invalid values and can be removed.
* **Correct wrong structure**: Values that don’t follow a defined structure can be removed. E.g. In a data set containing pin codes of Indian cities, a pin code of 12 digits would be an invalid value and needs to be removed. Similarly, a phone number of 12 digits would be an invalid value.
* **Validate internal rules**: If there are internal rules such as a date of a product’s delivery must definitely be after the date of the order, they should be correct and consistent.

Filtering Data:

After you have fixed the missing values, standardized the existing values, and fixed the invalid values, you would get to the last stage of data cleaning. Though you have a largely accurate data set by now, you might not need the entire data set for your analysis. It is important to understand what you need to infer from the data and then choose the relevant parts of the data set for your analysis. Thus, you need to filter the data to get what you need for your analysis.



Let’s summarise what you learnt about filtering data. You could use this as a checklist for future data cleaning exercises.

* **Deduplicate data**: Remove identical rows, remove rows where some columns are identical
* **Filter rows**: Filter by segment, filter by date period to get only the rows relevant to the analysis
* **Filter columns**: Pick columns relevant to the analysis
* **Aggregate data**: Group by required keys, aggregate the rest

Right Outer join

**Feedback :***The right outer join would keep all Employee IDs from the right table, and add columns from first table for IDs that are common to both the data sets. The question requires a merge that returns all unique rows from both the tables.*

Full outer join

**Feedback :***Full outer join is used to merge the data sets such that, apart from keeping the common Employee IDs, it also includes the Employee IDs unique to both the data sets.*