DATA PREPARATION
CLEANING DATA

Paul Speaker

Recoding Data in SQL

- One last tool from SQL: recoding data
- Recoding data is transforming the data values based on rules
- Has a format similar to “if…then”, but instead the keyword is CASE with conditionals
- Conditional for missing → IS NULL
- Example:

```
rules <- sqldf("SELECT CASE
        WHEN quantity > 10 THEN 'Large’
        WHEN quantity <= 10 AND quantity > 5 THEN 'Medium’
        ELSE 'Small' END
        AS rule
        FROM order_details")
```

Cleaning the Data

- Now that you have the Y and X data, it is time to start cleaning the data
  - Note: you generally do not want to clean data before it is all together
- You have look at the data to find problems with the data
- What data cleaning is not:
  - Getting rid of data you do not like
- When you have data you suspect as not "clean" you have 3 choices
  - Keep
  - Modify
  - Remove
- When you clean data, do not overwrite the original raw data
  - good for checking

Cleaning Data

- Structural problems with data are the best reason to clean it
  - “Data is weird” is not a reason to get rid of it
- There is no exhaustive list of why data needs to be cleaned. But important some examples
  - Missing data
  - Data that has the “wrong” kind of values
    - Negative values where values should be positive
    - Categorical values not in range
  - Incompatible data
    - Example: data scaled differently in the same field
  - Data that has an incorrect timestamp
    - 1/1/1900
    - Complaint data from 2008 when customer was only a customer from 2010-2020

Duplicate Data

- Datasets can have rows copied many times
- Some reasons
  - Incorrect joins (many joins are supposed to be 1-1, but aren't)
  - Data appended from multiple sources
- When checking for duplicate rows, check all rows in raw data, not just the rows in data you ultimately use
  - A good use of id and date fields!
- In R, there are a couple of functions that will remove duplicate rows
  - unique()
  - distinct()

EDA for Cleaning Data—Missing Values

- Discovering missing values is important
- Missing values can appear in several different ways
  - NA (true missing value)
  - “” (character field)
  - 0 (numeric field that does not make sense to be 0)
- How to discover
  - For NA’s count NA’s➔is.na(field)
  - In general, you can use our trick sum(conditional) to count the missing values
- How to handle
  - Why is it missing?
  - Would live data have missing values?
  - You won’t know until you investigate (domain expertise)
  - If it could be missing in live data, you have to keep
    - Might recategorize however (see next slide)

Handling Missing Values

- How to handle depends on source, use, purpose, and type
- If the missing data is the target, have to throw out
- If future live data will see missing values, it has to be kept
- If the field is numerical
  - Does it make sense to recode it to 0?
  - Will you miss those missing rows?
- If the field is categorical
  - Can missing be a separate category? (usually the right answer)
- Alternative: build 2 models
  - One with non-missing data, using the x
  - One with missing data, not using the x

EDA for Cleaning Data—Structural Issues

- Numerical targets that can only be positive often have negative or zero values
- Categorical fields that only have “Y” or “N” responses might have “X” has an answer
- How to handle—Investigate!
- For categorical, nothing wrong with leaving as is
- For numerical, there can be a lot going on
- Again, ask where data came from and what live data will model see
- Sometimes numerical data can be transformed to make it consistent
  - Different units
  - Different financial records of the same order