# DATA TIDYING

Applied Analytics: Frameworks and Methods 1

## Outline

- In this section, we will review Data Tidying approaches including
  - *Parsing*
  - *Variable Transformation*
  - *Restructuring*
  - *Joining tables*
  - *Addressing missing data*

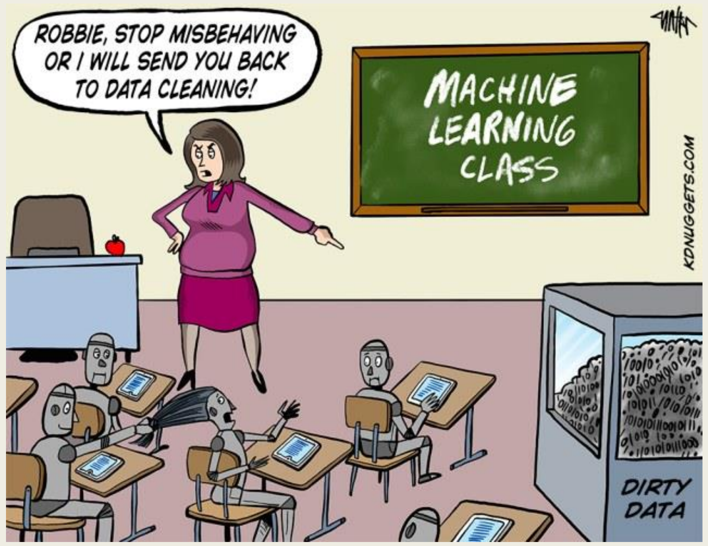
## Data Tidying

- Data seldom comes in a form that is ready for analysis. This may be because
  - *Variables are not in the correct format*
  - *Variables are too raw for analysis*
  - *Structure of data is not amenable for analysis*
  - *Data is spread across multiple tables*
  - *Some of the data is missing*

## Data Tidying

- Data tidying is the process of transforming the data from one form to another to make it more appropriate for downstream analysis.
- It is one the less celebrated, yet critically important tasks of a data scientist
- Data tidying will often consume 80% of the time for a data science project.
- Data tidying goes by many names including

| | | |
|---|---|---|
| Munging | Cleaning | Carpentry |
| Manipulation | Transformation | Wrangling |


ROBBIE, STOP MISBEHAVING OR I WILL SEND YOU BACK TO DATA CLEANING!
MACHINE LEARNING CLASS
KDNUGGETS.COM
DIRTY DATA

## Data Tidying

- We will examine the following methods to make the data analysis ready
  - *Parsing: When variables are not the correct format*
  - *Transformation: When variables are too raw for analysis*
  - *Restructuring: When the structure of data is not amenable for analysis*
  - *Joining tables: When data is spread across multiple tables*
  - *Imputation and Deletion: When data is missing*

## PARSING

## Parsing

- Data imported must assign the correct class to variables:
  - *Numbers using a class like integer or numeric*
  - *String as character*
  - *Boolean as logical*
  - *Dates using a Date class*
  - *Categorical variables as factor and so on*

## Parsing

- Numbers
  - *Conventions for writing numbers differ across the world. For e.g., in the US, we use a period for a decimal point and a comma for grouping numbers but France does the exact opposite.*
  - *Numbers may include characters or words to show context. E.g., $25, 25%, 25% increase in wages*

## Parsing

- Words
  - *As such, words are easy to parse.*
  - *But this simple task is complicated by the use of special symbols (e.g., Häagen-Dazs), non-english languages, and emojis(e.g., 💩).*
  - *The ASCII system of encoding characters was not designed for such use cases.*
  - *This has given way to UTF-8, a more expansive and universal character coding format that extends the lexicon of ASCII.*

## Parsing

- Dates
  - *The conventions for expressing date and time vary across the world.. For e.g., US Independence Day can be written as*
    - July 4, 1776
    - 4/7/1776
    - 7/4/1776
    - 1776/7/4

## Illustration

See DataParsing.html

## VARIABLE TRANSFORMATION

## Variable Transformation

- Variables contained in a dataset seldom come in an analysis-ready format.
- Certain variables may need to be transformed before they can be used for predictive analysis.
- The nature and types of transformation depend on whether the variable is categorical or numeric.
- Examining all variable transformations is beyond the scope of this course, but we will examine commonly used transformations.

## Transformations
## For Categorical Variables

- Dummy Variables: A categorical variable with k levels is represented using k-1 dummy variables, where one of the levels serves as a reference level. A dummy variable can only take on two values, 0 and 1.
- Remove Low Variance Variables: Remove variables with zero variance or low variance.
- Lump categories: Combine sparse categories with a related category
- Other Category: Combine rare categories into an Other category

## Transformations For Numeric Variables

- Scaling: Place all predictors in the same units by methods such as centering, standardizing and range scaling.
- Skewness: Apply a functional transformation (e.g., log, inverse) to address skewness.
- Low Variance: Remove variables with zero variance or low variance.
- Combine Variables: Construct a weighted or unweighted index by combing variables.
- Binning: Group a numeric variable into a set of two more discrete bins.

## Illustration

See VariableTransformation.html

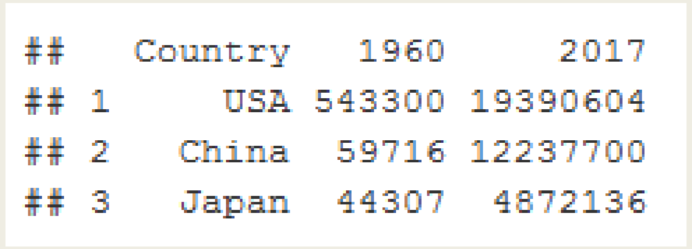
## DATA RESTRUCTURING

## Restructure Data

- Sometimes, structure of the data may get in the way of analyses
- Hadley Wickham recommends the use of a *tidy* data structure which is one in which
  - *Each variable has its own column*
  - *Each observation has its own row*
  - *Each values has its own cell*
- Choice of structure is often driven by the functions to be used and analytical technique to be conducted.

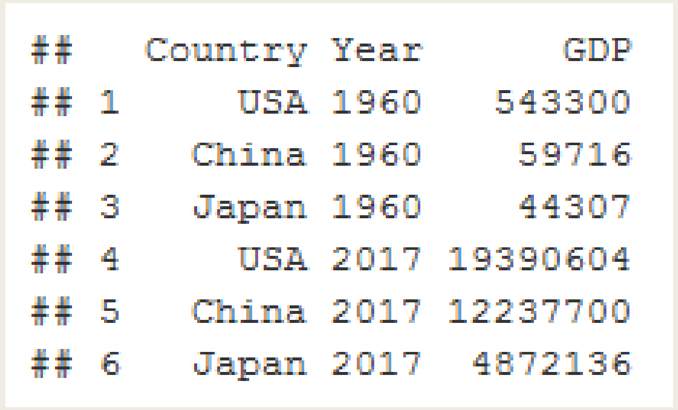
## Tidy Data

### Wide Data



```
##    Country   1960     2017
## 1      USA 543300 19390604
## 2    China  59716 12237700
## 3    Japan  44307  4872136
```

### Tall Data



```
##    Country Year      GDP
## 1      USA 1960   543300
## 2    China 1960    59716
## 3    Japan 1960    44307
## 4      USA 2017 19390604
## 5    China 2017 12237700
## 6    Japan 2017  4872136
```

## Tidy Data in R

- gather() or pivot_longer():
  - *Convert wide data to tall format*
  - *Most functions and software prefer expect data to be in tall format*
- spread() or pivot_wider()
  - *Convert tall data to wide format*

## Illustration

See DataStructure.html

## JOINING

## Joins

- Data is generally spread across multiple tables rather than in a single flat file.
- These tables share a common variable or field, thus are related (which is why these tables are also known as relations). All the tables or relations together comprise a relational database.
- Storing data across relations offers a number of benefits such as reduced redundancy, fewer errors from updates, and greater efficiency in storage and retrieval.
- The data required for analysis at hand is retrieved by linking relations through joins. The languages most widely used for this is SQL, however, this can also be accomplished using R.

# Joins

- Types of Joins
  - *Inner Join*
  - *Outer Joins*
    - Left Outer Join
    - Right Outer Join
    - Full Outer Join

## Illustration

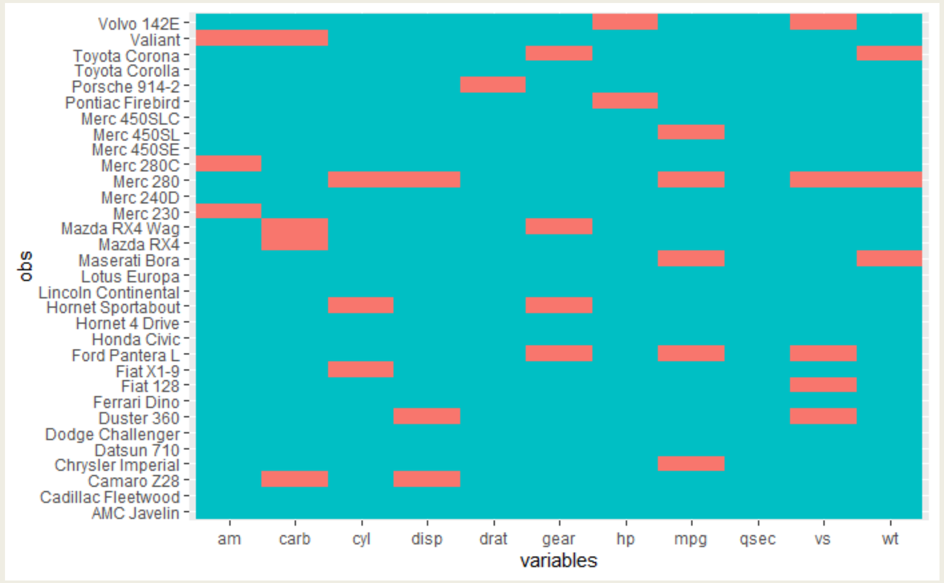
See Joins.html

## MISSING DATA

is.na(NA)

# Missing Data

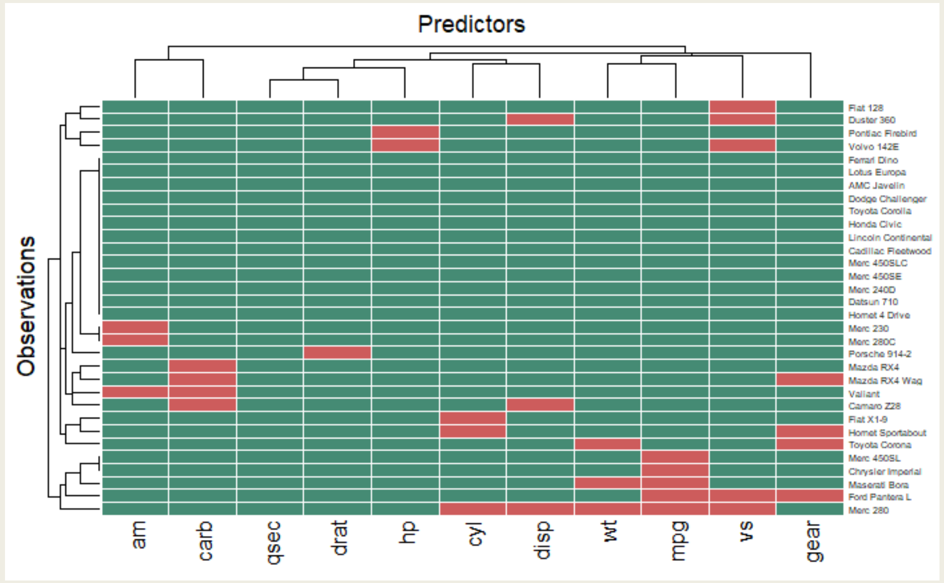
- Common in real datasets
- Critical to address it as many predictive models require complete data
- Important to understand reasons for missing data in order to decide what do with it



# Missing Data

- Possible reasons may be
  - *Random*
  - *Data deficiency*
  - *Specific causes*

- Pattern in a heatmap may highlight reasons



## Solutions

1. Delete
2. Encode Missing Data
3. Impute

- Simplest solution. Delete entire row if any value is missing
- However,
  - *small number of missing values can lead to losing a large percent of sample.*
  - *acquiring new data may be costly*
  - *potential of introducing bias*

## Solutions

1. Delete
2. Encode Missing Data
3. Impute

- Hard code missingness
- In a categorical variable, missing data may be encoded as another level
- Missing data in a continuous variable may be represented using a dummy variable.

## Solutions

1. Delete
2. Encode Missing Data
3. Impute

- Information and relationships among non-missing predictors is used to estimate missing values
- Impute no more than 20% missing data for any variable
- Conducted prior to other steps in data preparation
- Methods include predictive mean matching, kNN, trees, tree-based methods
- R Packages include mice, caret

## Illustration

See MissingData.html

## Conclusion

- In this module, we learnt
  - *ways to parse data*
  - *how to transform variables in raw data to useful inputs*
  - *how to restructure a dataset*
  - *the concept of relations and how to join tables*
  - *methods for addressing missing data*