CDS

Cornell Data Science

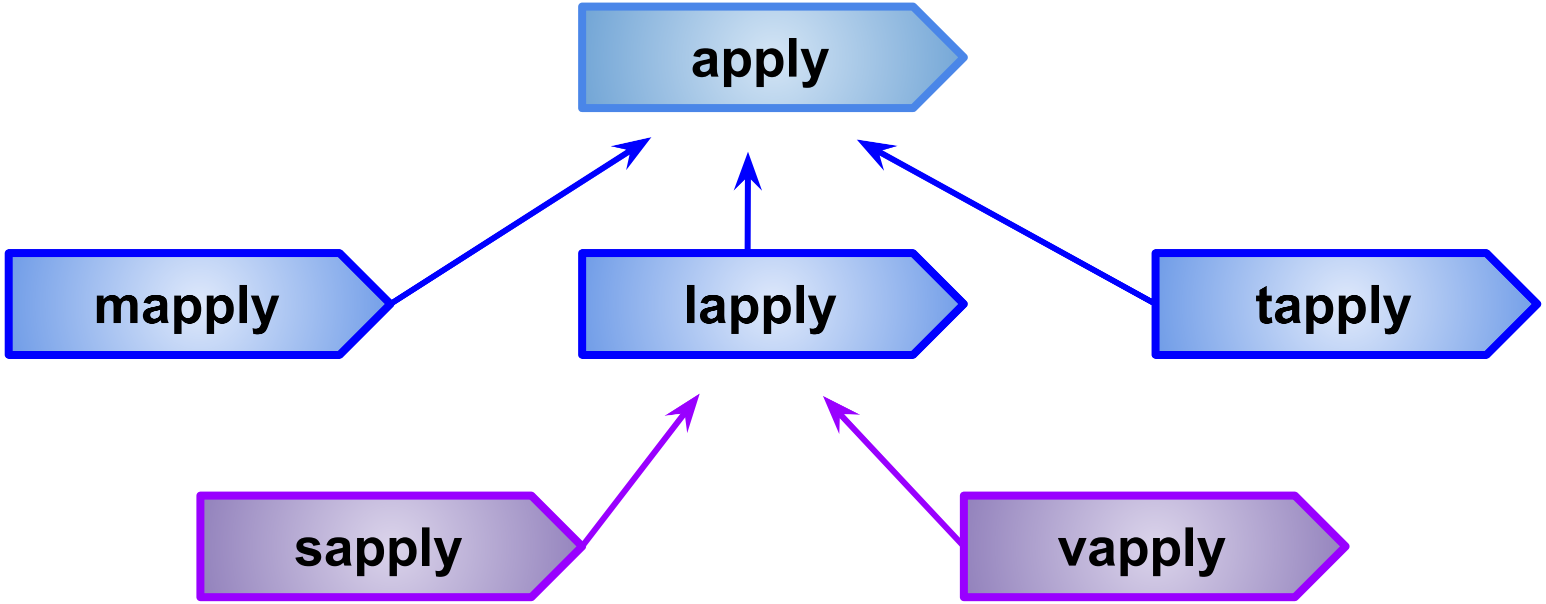
Data Manipulation

$$ Golden Rules of Writing Fast R $$

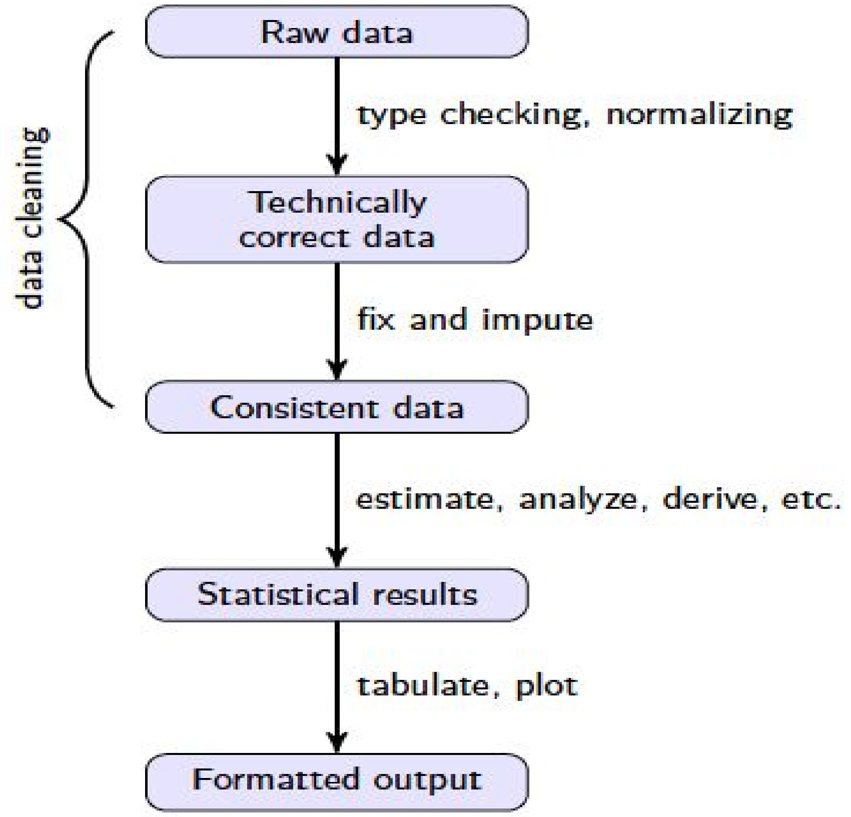
Vectorize your operations as much as possible

Minimize the number of vectors to operate on

The Apply Family



The Data Pipeline



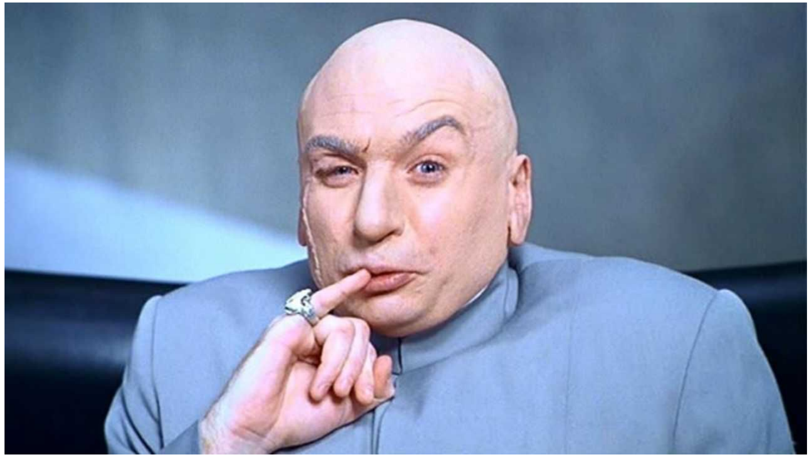
Source

Question:

What are some ways in which data can be “messy”?

Data Manipulation Tools

1. R base functions
2. `plyr` package
3. `dplyr` package
4. `sqldf` - for SQL users
5. Data.table package



Source

Manipulation Tool #1: Filtering

What it does: Grabs a subset of the rows in a data frame with a condition

Name	**Age**	**Major**
Amit	19	Computer Science
Dae Won	24	ORIE
Chase	19	Information Science
Jared	19	Computer Science
Kenta	20	Computer Science

Manipulation Tool #2: Subsetting

What it does: Grabs a subset of the columns in a data frame.

Name	Age	Major
Amit	19	Computer Science
Dae Won	24	ORIE
Chase	19	Information Science
Jared	19	Computer Science
Kenta	20	Computer Science

Manipulation Tool #3: Combining

What it does: Joins together two data frames, either row-wise or column-wise.
How to do it: `cbind` and `rbind` operators in R.

Name
Amit
Dae Won
Chase
Jared
Kenta

Age	Major
19	Computer Science
24	ORIE
19	Information Science
19	Computer Science
20	Computer Science

Name	Age	Major
Amit	19	Computer Science
Dae Won	24	ORIE
Chase	19	Information Science
Jared	19	Computer Science
Kenta	20	Computer Science

Manipulation Tool #4: Joining

What it does: Joins together two data frames, combining rows that have the same value for a column.

How to do it: Use dplyr's join method or the `merge` and `cbind` operators in R.

Manipulation Tool #5: Summarizing

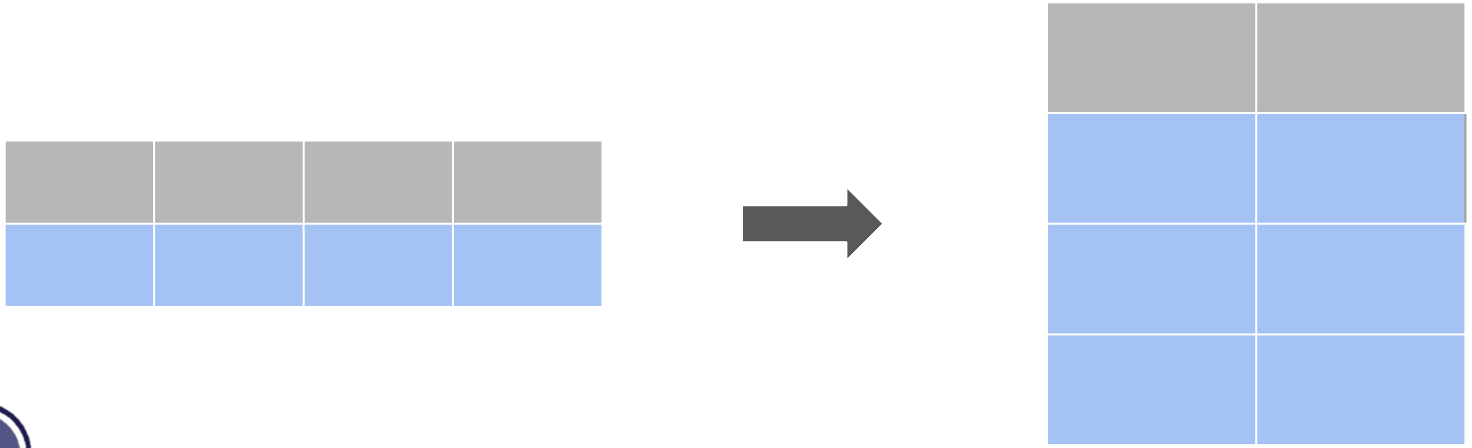
What it does: Computes aggregate data about the data frame, such as the number of elements with a given property.

How to do it: Use dplyr’s summarize and aggregate methods.

tidyr Library

What it does: Convert Wide data.frame to Longer data.frame

How to do it: gather( ), spread( ), unite( ), separate( )



Column Manipulation 1

spread()

```
spread(data, Year, GPA)
```

gather()

```
gather(data, Year, GPA, Freshman:Junior)
```

Name	Year	GPA
Hermione	Freshman	4.2
Hermione	Sophomore	4.3
Hermione	Junior	4.1

↕

Name	Freshman	Sophomore	Junior
Hermione	4.2	4.3	4.1

Column Manipulation 2

separate( )

```
separate(data, TA, c("Name", "Major"), sep = ": ")
```

unite( )

```
unite(data, “TA”, c(Name, Major), sep = ": ")
```

TA
Amit: CS
Dae Won: OR
Chase: IS
Jared: CS
Kenta: CS

Name	Major
Amit	CS
Dae Won	OR
Chase	IS
Jared	CS
Kenta	CS

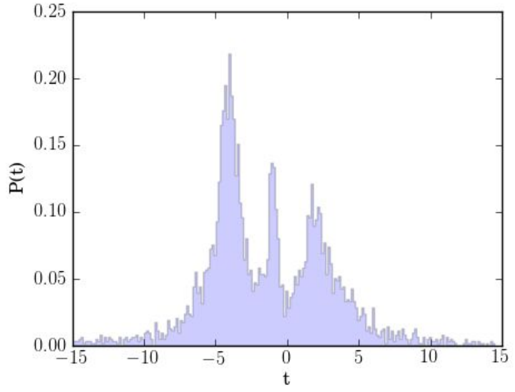
Using the Force

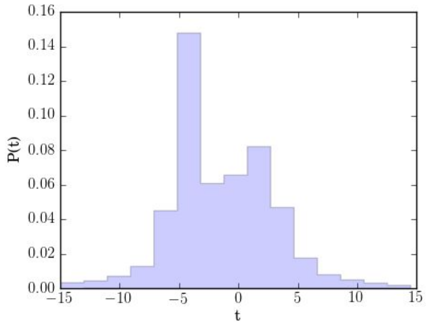
Source

Technique #1: Binning

What it does: Makes continuous data categorical by lumping ranges of data into discrete "levels."

How to do it: Create a new categorical variable. Assign each category using logical vectors on the numerical column.





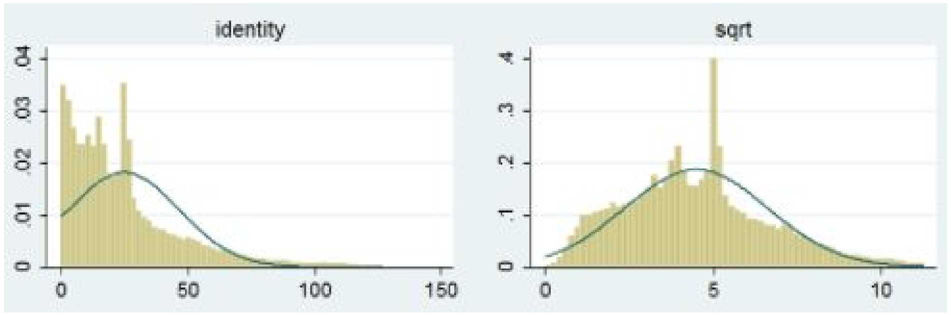
Source

Technique #2: Normalizing

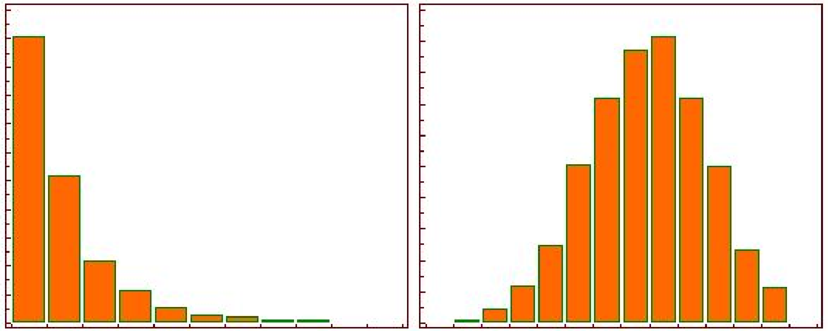
What it does: Turns the data into a bell curve, or Gaussian, shape.

How to do it: Apply a function, or transformation, to each column (`dplyr` mutate).

Square root transformation



Log transformation



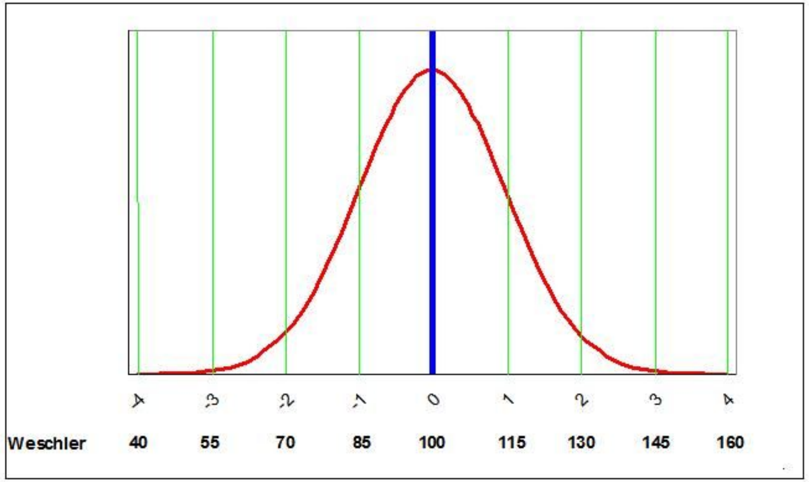
Others include cubic root, reciprocal, square, cube…

Source Source

Technique #3: Standardizing

What it does: Centers the data with mean 0 and variance 1.

How to do it: Subtract the mean from each data point and divide by the standard deviation. Use the `mean` and `sd` functions.



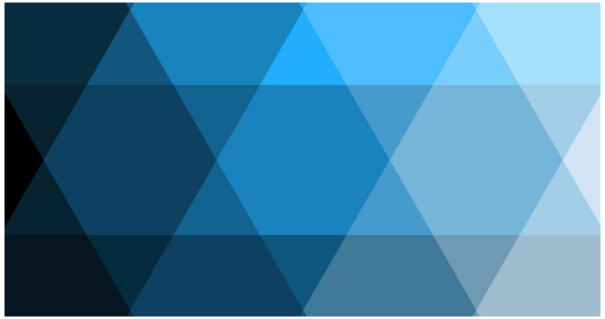
Source

Technique #4: Ordering

What it does: Converts categorical data that is inherently ordered into a numerical scale.

How to do it: Assign values using logical vectors.

```
students$Age[which(students$Year==”SENIOR”)] <- 22
students$Age[which(students$Year==”JUNIOR”)] <- 21
# etc.
```



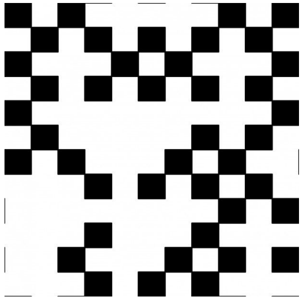
Source

Technique #5: Dummy Variables

What it does: Creates a binary variable for each category in a categorical variable.

How to do it: Create a new variable named after the category. Assign 1 to the variable when the categorical variable takes on the category.

```
students$Senior <- 0
students$Senior[which(students$Year==”SENIOR”)] <- 1
# etc.
```



Source

Coming Up

Your assignment: Assignment 2

Next week: Visualizing, animating, and presenting data

See you then!

Cornell Data Science