

# Data Cleaning & Exploration

RLadies

London, Ontario

Jaky Kueper

December 3, 2019

# Outline

1. Overview of data cleaning and exploration.
  2. Helpful packages and functions for 1.
  3. Practice and skill-share – hands on than last time!
- 
- Assumption: your data are rectangular and not super large, are not complete garbage (principles of data quality), and you have some sense of what's there.
  - Disclaimer: Jaky studies epidemiology and computer science; she is a self-taught R-enthusiast.

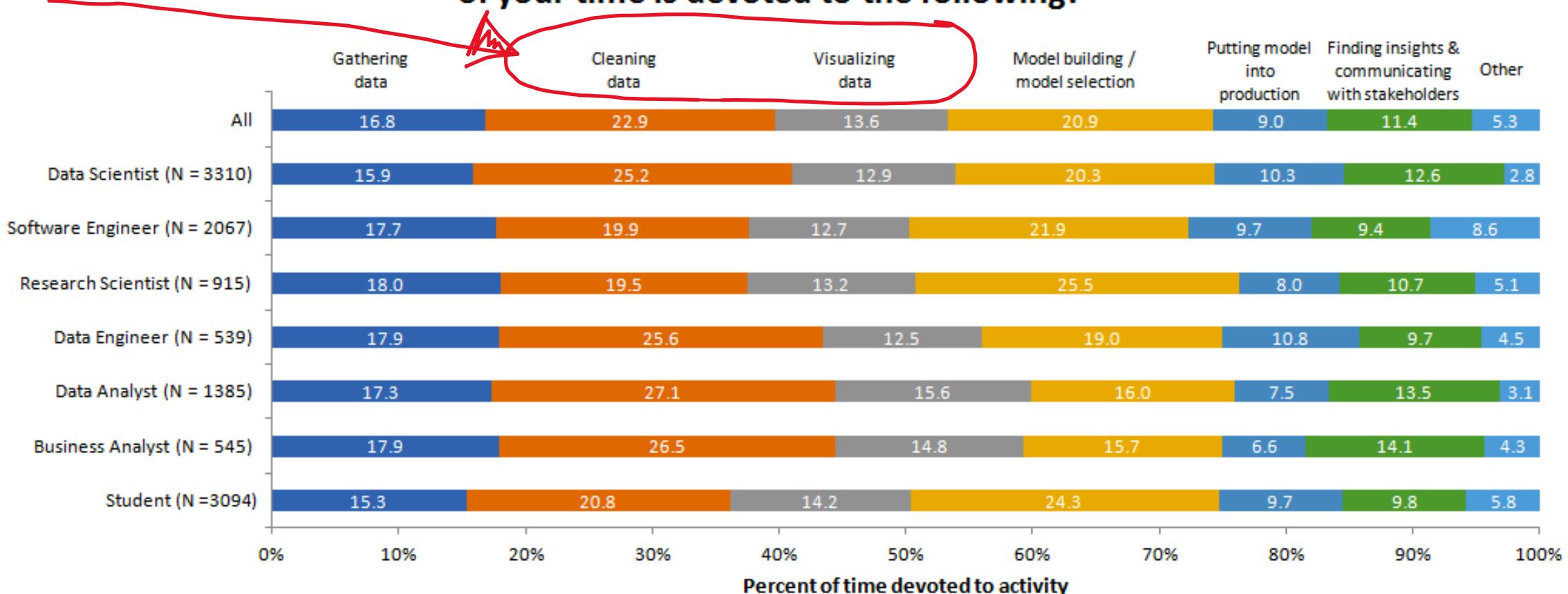
# Introduction to Data Cleaning



# Data Cleaning and Exploration

- **Data Cleaning** is “the process of amending or removing data in a database that is incorrect, incomplete, improperly formatted, or duplicated.” (Margaret Rouse)
  - AKA ‘tidying’ or ‘wrangling’ = ‘tidy’ + ‘transformation’ (Wickham & Golemud)
  - AKA data cleansing
- **Data Exploration** is “the first step in data analysis and typically involves summarizing the main characteristics of a data set, including its size, accuracy, initial patterns in the data and other attributes.” (Margaret Rouse)
  - Note: we are not getting into things like exploratory risk factor analysis or descriptive epidemiology.
- What these entail depends on the data and what you want to do.
- How well you prepare data impacts quality of results!

**During a typical data science project at work or school, approximately what proportion of your time is devoted to the following?**



Note: Data are from the 2018 Kaggle ML and Data Science Survey. You can learn more about the study here: <http://www.kaggle.com/kaggle/kaggle-survey-2018>. A total of 23859 respondents completed the survey; the percentages in the graph are based on a total of 15937 respondents who provided an answer to this question. Only selected job titles are presented.

# Tidy Data

country	year	cases	population
Afghanistan	1990	745	1857071
Afghanistan	2000	2666	2095360
Brazil	1999	31737	17206362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	216766	128042583

variables

country	year	cases	population
Afghanistan	1990	745	1857071
Afghanistan	2000	2666	2095360
Brazil	1999	31737	17206362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	216766	128042583

observations

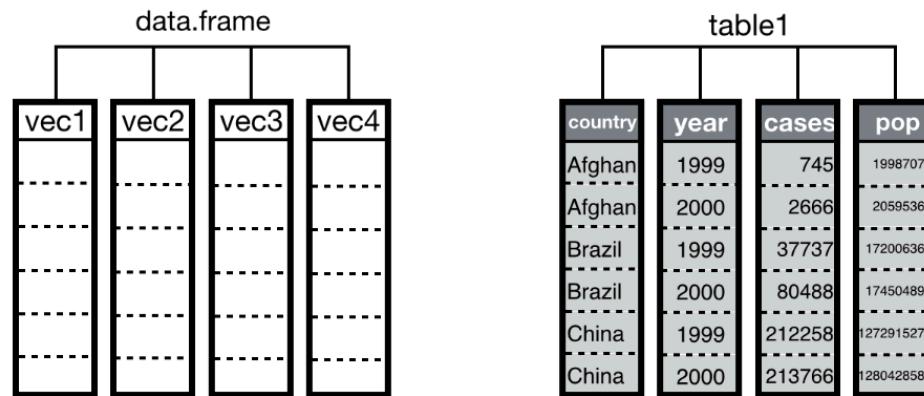
country	year	cases	population
Afghanistan	1990	745	1857071
Afghanistan	2000	2666	2095360
Brazil	1999	31737	17206362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	216766	128042583

values

- Consistently display values AND relationships.

<https://garrettgman.github.io/tidying/>

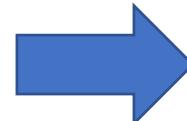
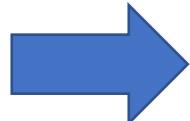
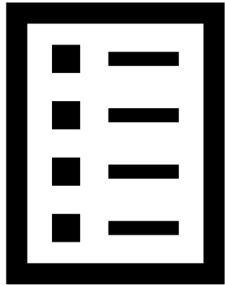
# Tidy Data



*A data frame is a list of vectors that R displays as a table. When your data is tidy, the values of each variable fall in their own column vector.*

- Best format for R's functions – saves time and energy later!

# Sample Workflow



### 1. Make a plan

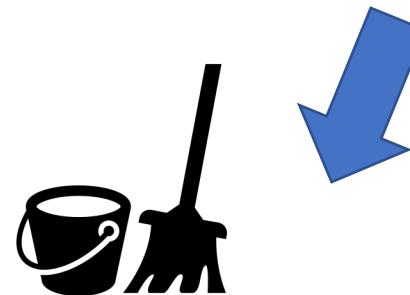
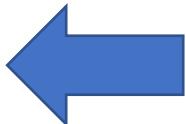
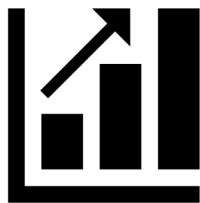
- Outline what you want to do
- Don't avoid pen & paper.

### 2. Explore the data

- Get to know the dataset(s).
- Note changes to be made & questions.

### 3. Gather Info to resolve uncertainties

- Ask questions!
- Research.
- Further explore the data.



### 6. Analyze!

- Data are ready to use.

### 5. Verify

- Re-explore to check distributions, format, etc.

### 4. Get the data into a usable format

- Includes modifying or creating new columns.
- Create a reusable script.

# A gift to your future self

- Create a script (or defined part of a script) that gets your data into ‘working order’.
  - Hit run and go.
  - Comment to remember what you did (or did not do) and why!
  - Promotes reproducibility and collaboration.
- Make a copy of your data frame or set it up to easily be reloaded.
- Recycle functions or general processes for future projects.

# 1. Make a Plan

# Before diving into the data...

- Outline what you want to do
  - Goals and sub-goals
- Sample questions
  - Variables needed
  - Observations needed
  - The form of data (wide versus long)

## 2. Explore the Data

Our goal here is to get a sense of the data, not to make beautiful plots.

# Initial Exploration

- Explore dataset characteristics
  - Size
  - Shape
  - Contents
  - Data types
  - Keys for relational databases
  - Sample observations

`df; View(df); names(df); glimpse(df); str(df)`  
`dim(df); nrow(df); ncol(df); length(df$col)`  
*Wide: columns usually represent groups.*  
`Summary(df); summary(df$colName)`  
`Typeof(df$column)`  
`df %>% count(primaryKey) %>% filter(n > 1)`  
`Head(df); tail(df); df[c(3:5), ]; df[“rowName”, ]`

For a review of datatypes:

<https://swcarpentry.github.io/r-novice-inflammation/13-supply-data-structures/>

# Initial Exploration

- Explore dataset contents
  - Summary statistics / data profiling
- Questions to ask yourself
  - Anything unexpected?
  - Are any known values correct?
    - *Mean, median, range, mode, counts, etc.*
- Base functions to get a quick overview
  - *Summary(df)*
  - *Summary(df\$numericColumn)*
  - *Table(df\$characterColumn)*
- You may prefer to do column renaming or other tidying before this step.

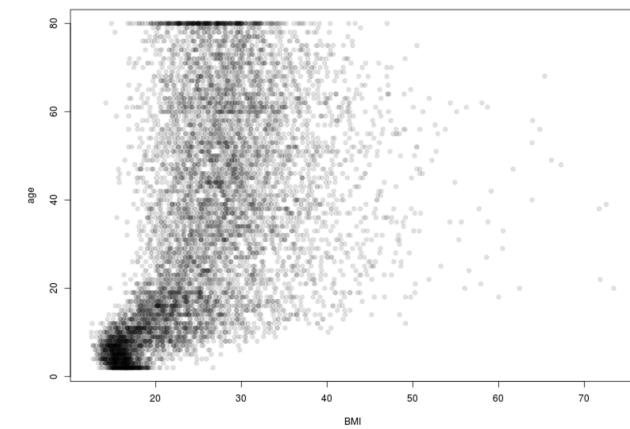
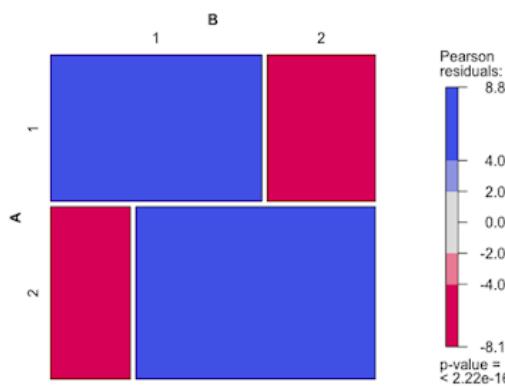
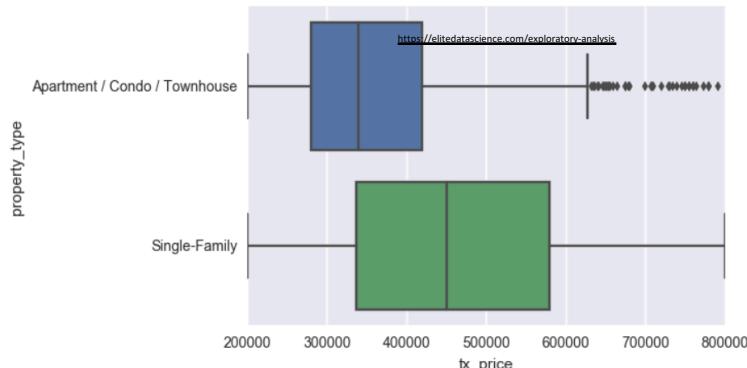
# Initial Exploration: Within Column Plots

- Numerical distributions
  - Histograms
    - `Hist(df$numericColumn)`
  - Watch out for: Unexpected trends, ranges, outliers, scale/sub-ideal data types, etc.
- Categorical distributions
  - Bar plots
    - `barplot(vectororMatrixWithNumericCounts); barplot(table(df$characterColumn))`
  - Watch out for: small categories, categories to collapse, ordering, trends, etc.

Note: sometimes it will make more sense to do this after cleaning up data columns.

# Initial Exploration: Between Column Plots

- Categorical by numerical distributions
  - Box plots
    - `Boxplot(formula, df)`
- Numerical by numerical
  - Correlations
    - `Cor()`
  - Scatterplot
    - `Plot(df$var1, df$var2)`
- Categorical by categorical
  - Mosaic plot
    - `Mosaicplot(table(df$var1, df$var2))`



Plot tutorial:  
<https://datawookie.netlify.com/blog/2013/05/plotting-categorical-variables/>

# 3. Gather Information

# Ask/Inquire

- Primary vs. secondary data
- Questions may be about anything you require to better understand and use the data!
  - Where the data come from.
  - Data definitions/dictionary.
  - Standard practices, e.g. units.
- Goal: to feel confident in decisions about how to manipulate the data going forward.
  - May require many consultations and looking for data in different places.

# Example: Electronic Health record ‘level of education’ field

Field value	Actual Meaning
“ ”	
Do not know	
Prefer not to answer	
Null	
Unknown	
Undefined	
Other	
Defined value	

# Example: Electronic Health record ‘level of education’ field

The value of consultation with people who generate or maintain the data!

Field value	Actual Meaning	
“ ”	Not asked	Mix of important and non-important differences here.
Do not know	Client was asked or given the opportunity to provide info	
Prefer not to answer	Client was asked or given the opportunity to provide info	
Null	Not asked	
Unknown	Not asked	
Undefined	Client provided information that did not fit a predefined category	
Other	Client provided information that did not fit a predefined category	
Defined value	As stated (4 levels) 'No formal education' to 'Post secondary or equivalent'	

# Example: Electronic Health record ‘level of education’ field

It is important to consult with people who generate or maintain the data

Field value	Actual Meaning
“ ”	Not asked
Do not know	Client was asked or given the opportunity to provide info
Prefer not to answer	Client was asked or given the opportunity to provide info
Null	Not asked
Unknown	Not asked
Undefined	Client provided information that did not fit a predefined category
Other	Client provided information that did not fit a predefined category
Defined value	As stated (4 levels) 'No formal education' to 'Post secondary or equivalent'

Treating all these the same may introduce bias.  
(Not wanting to tell a care provider about your education is different than them never asking you about it.)

# Example: Electronic Health record ‘level of education’ field

How these values are treated could change results.

Field value	Actual Meaning
“ ”	Not asked
Do not know	Client was asked or given the opportunity to provide info
Prefer not to answer	Client was asked or given the opportunity to provide info
Null	Not asked
Unknown	Not asked
Undefined	Client provided information that did not fit a predefined category
Other	Client provided information that did not fit a predefined category
Defined value	As stated (4 levels) 'No formal education' to 'Post secondary or equivalent'

Treating these differently may introduce bias

(Values are based on care provider behaviour and location / the type of EMR being used.)

# 4. Cleaning

# Remove Irrelevant and Unwanted Data

- Remove columns – complete variable removal
  - Declutter (e.g. unneeded, duplicate, highly correlated (unless using for e.g. missing data)).
  - `Dplyr::select()` can specify what to keep or remove.
- Remove rows
  - Duplicate observations, e.g. if joined datasets
  - Unwanted observations, e.g. to refine target population
  - Outliers should only be removed if you have a good reason!
  - `Dplyr::filter()` keeps a subset of rows in a column.

# Remodel

- Combine or recode analogous categorical variables
  - E.g. syntactic fixes (capitalization, spelling, synonyms), domain-knowledge decisions (similar categories), small categories (Keep? Remove? Combine?)
  - E.g. `dplyr::arrange()` to alter ordering
- Work with strings (bunch of characters)
  - E.g. search and replace.
- Missing data strategy
  - Remember missingness can be informative.
- Standardize
  - Rename variables
    - camelCaseIsMyFavourite
    - Avoid\_spaces\_and\_most\_nonalphanum\_characters
  - Consistent units for numerical variables
- Convert between datatypes
  - `is.type()` gives True/False
  - `as.type()` converts it

# Create (usually new columns)

- Calculate new variables
  - E.g. `percentOfPopulation = cases / population * 100`
- Dummy variables
  - Each level of a categorical variable becomes a new column with values 0 or 1.
- Aggregate values
  - E.g. Diagnostic codes to identify everyone with a particular health condition.
- Note you may want to remove columns that become irrelevant.

# 5. Verify

# Data Exploration

- Is everything ready to go?
- Can use similar tools as initial data exploration.
  - May want prettier presentation...Check out previous meetup materials!



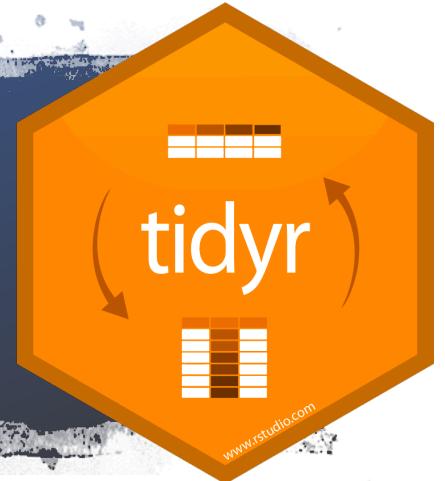
# 6. Analyze

For another day!

# Packages & Functions



# Tidyr: reshape data



Long

country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2

Wide

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

`pivot_longer(table2, key, value)`

`pivot_longer(df, "keyColumnName", "ValueColumnName", ...)`

Distribute key:value columns into data cells.  
(previously spread())

New key

country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

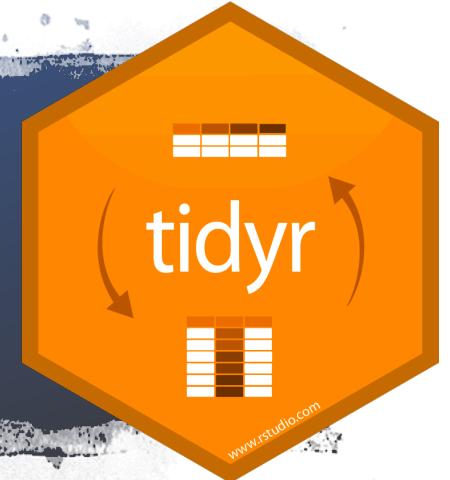
table4

`pivot_wider(table4, "year", "cases", 2:3)`

`Pivot_wider(df, "keyColumnName", "valueColumnName", columnsTo...)`

Collect column names and put them into a single 'key' column.  
(previously gather())

# Tidyr: reshape data



- Other useful functions: `separate()` and `unite()`
- Overview & cheatsheet: <https://tidyr.tidyverse.org/>
- Check out: <https://garrettgman.github.io/tidying/> or <http://www.milanor.net/blog/reshape-data-r-tidyr-vs-reshape2/> for comparison with `reshape2`

# Dplyr: data manipulation

- Verbs (we saw some earlier) :
  - filter
  - select
  - mutate
  - arrange
  - summarize
  - group\_by
  - Recode / rename
- Overwrite or save to a new dataframe.
- Overview & cheat sheet: <https://dplyr.tidyverse.org/>
- Tutorial with exercises: <https://datacarpentry.org/2015-07-22-JamesMadison/etc/r-datacarpentry.html>



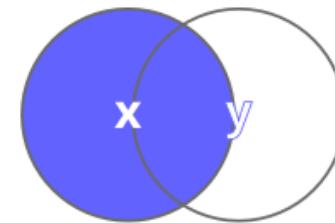


# Joins

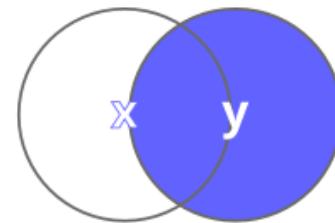
- With dplyr:  
[http://lindsaybrin.github.io/CREATE\\_R\\_Workshop/Lesson\\_-\\_dplyr\\_join.html](http://lindsaybrin.github.io/CREATE_R_Workshop/Lesson_-_dplyr_join.html)
- Relational databases:  
<https://r4ds.had.co.nz/relational-data.html>

## dplyr joins

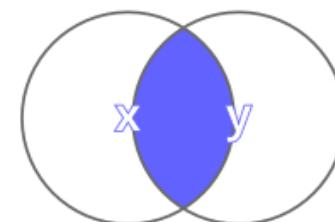
`left_join(x, y)`



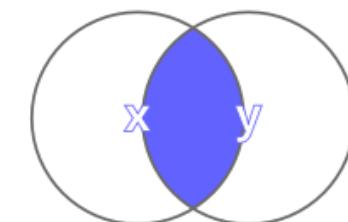
`right_join(x, y)`



`inner_join(x, y)`

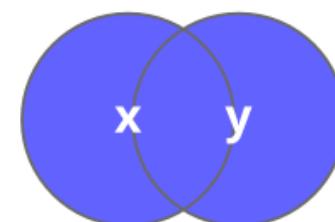


`semi_join(x, y)`

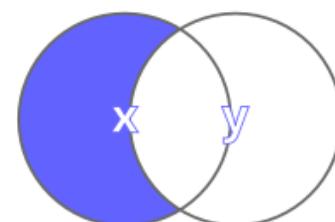


(never duplicate rows of x)

`full_join(x, y)`



`anti_join(x, y)`

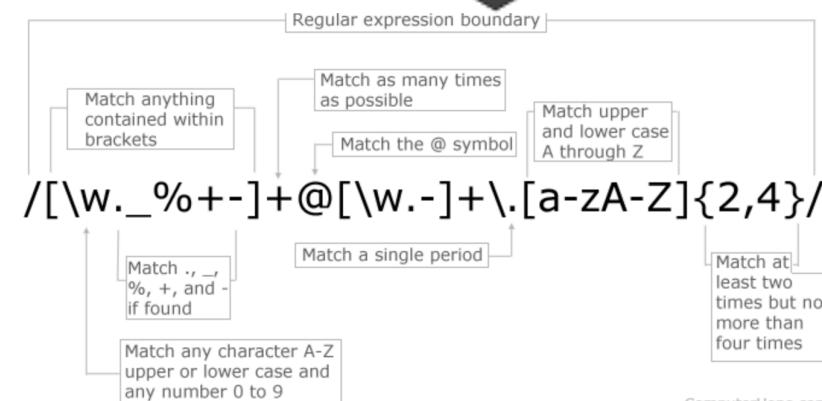




# Stringr: working with strings



- Strings are series of characters.
  - Create using ` ` or ` `
- Stringr helps you match, split, and manipulate character data.
  - Relies on regular expressions
- Can use to ‘tidy’ character variable types.
  - E.g. apply a stringr function to each row in a character variable column.
  - For loops or functions can be helpful.
- Overview & cheatsheet:  
<https://www.rdocumentation.org/packages/stringr/versions/1.4.0>



ComputerHope.com

# magrittR: code readability



- Primary operator is the pipe: `%>%`
  - 'evaluate the left hand side, and feed the result as input to the function on the right hand side'
- Main advantage: code readability
  - Use with things like subsetting, arranging, and mutating data.
- Overview: <https://magrittr.tidyverse.org/>
- Tutorial with dplyr: <https://seananderson.ca/2014/09/13/dplyr-intro/>

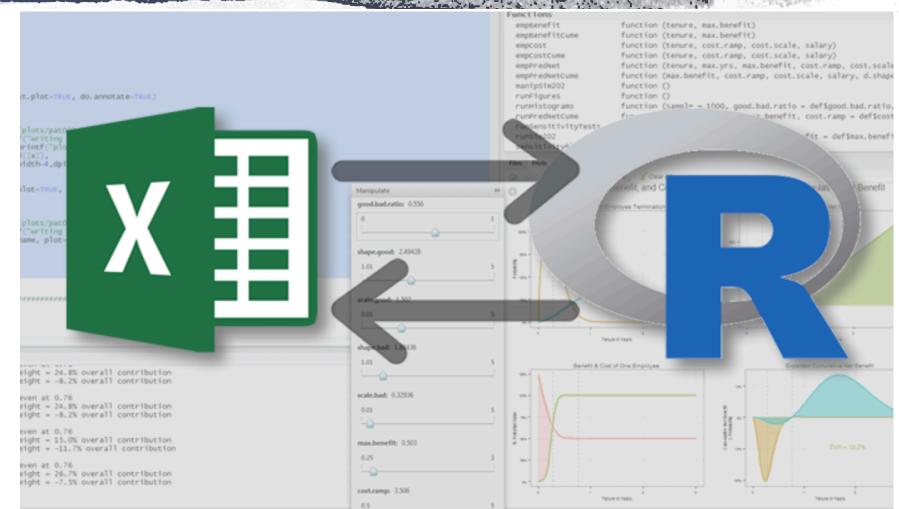
# Lubridate: dates and time



- Overview & cheat sheet: <https://lubridate.tidyverse.org/>
- How dates work in excel: <https://www.excelcampus.com/functions/how-dates-work-in-excel/>
  - Days since origin (the beginning of time: January 1, 1904 is common)
  - Reformatted to e.g. 05/30/2019

# Tidyxl: messy excel data

- Human readable → machine readable
- Resources:
  - <https://cran.r-project.org/web/packages/tidyxl/vignettes/tidyxl.html>
  - <https://github.com/nacnudus/tidyxl>



Possibly helpful dictionary for those transitioning from Excel to R:

<https://paulvanderlaken.com/2018/07/31/transitioning-from-excel-to-r-dictionary-of-common-functions/>

# Resources

- “Great R packages for data import, wrangling and visualization”
  - <https://www.computerworld.com/article/2921176/great-r-packages-for-data-import-wrangling-visualization.html>
- <https://garrettgman.github.io/tidying/>
- Cheatsheet:
  - <https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf>
- Helpful blog from Thea’s meetup:
  - <https://whattheyforgot.org/save-source.html>
- Table of useful R commands: <https://www.calvin.edu/~scofield/courses/m143/materials/RcmdsFromClass.pdf>

# Practice

Download the R Script and Data from the Rladies Meetup page!