Table Joins

by Marshall Markham

About Me

- Maxpoint Interactive
  - Located in Cary at RTP
  - Online marketing solutions
    - Ad serving and efficacy measurement
  - Roles
    - Business Analyst, Marketing Scientist, Staff Data Scientist
  - Stack
    - Python, PostgresSQL, HDFS , Spark

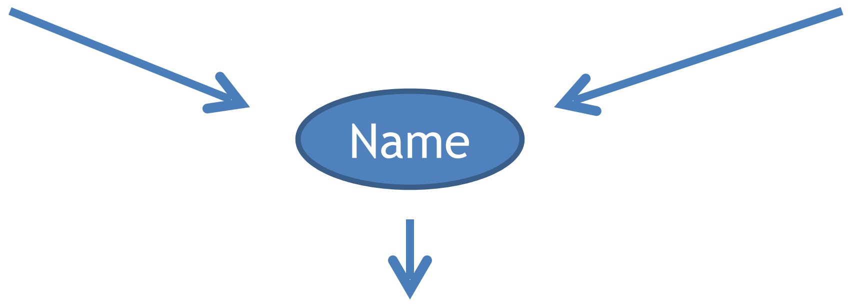
Joins

- Joins are a relational data tool that allow us to compare and/or combine separate datasets.
- Joins turn multiple tables into a single new table for analysis by comparing one or many of the columns.
- Joins are a concept implemented across languages.
- They are a fundamental concept in data science.

A Basic Join

Name	Age
Mary	25
John	29
Edna	63

Name	Height (In)
Mary	77
John	68
Edna	60



Name	Age	Height (In)
Mary	25	77
John	29	68
Edna	63	60

A Database

- DB
  - Configurations
  - Locations
  - Ad Serving
  - Demographics
  - Third Party Reporting

A Database at the Table Level

- DB
  - Configurations
    - Campaigns
    - Sub Campaigns
  - Locations
    - Zip code
    - State
  - Ad Serving
    - Campaign performance
    - Campaign performance by geo
  - Demographics
    - Demographics
  - Third Party Reporting
    - Blocked ads
    - Externally monitored performance

A Database at the Column Level

- DB
  - Configurations
    - Campaigns
      - Campaign Id, Name, Business Analyst, …
    - Sub Campaigns
      - Campaign Id, Subid, Planned Impressions, Planned Budget, Start, End, …
  - Locations
    - Zip code
      - Zip, State, Region size, Center lat, Center lon, ….
    - State
      - Name, Abbreviation, Size, ….
  - Ad Serving
    - Campaign performance
      - Subid, Clicks, Impressions Served, Timestamp, …
    - Campaign performance by geo
      - Subid, Clicks, Zip, Impressions Served, Timestamp, …
  - Demographics
    - Demographics
      - Zip, Pct Bachelors, Pct Asian, Pct Income Greater than 100k, …
  - Third Party Reporting
    - Blocked ads
      - Subid, Impressions Counted, Timestamp, …
    - Externally monitored performance
      - Subid, Events, …

A Quote from My Mentor

- Often Data Science comes down to feature engineering.
  - Marius Van Niekerk

In Context

- Feature Engineering:
  - The process of finding predictors and responses on which to do our data science.
- Joins allow us to place any newly derived/ found data next to the data we already have.

Feature Engineering

- Can we find performance by campaign?
- Can we find performance by education level?
- Can we find impression loss by geography?

The Three Standard Joins

- Inner
- Outer
- One Sided Outer
  - Left or Right

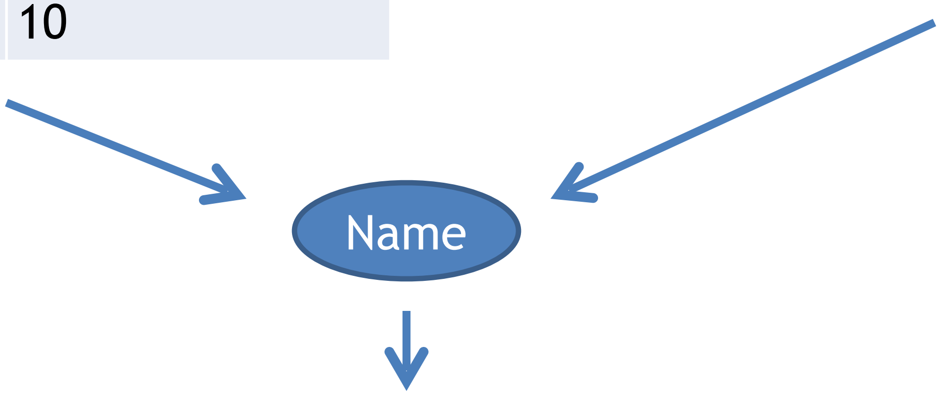
Join Rules

- Inner
  - Create a row for each match.
- Left Outer (Right Outer is similar)
  - Create a row for each match. For each left table row which is unmatched, create a row placing NULL values in the right table columns.
- Outer (aka Full Outer)
  - Create a row for each match. For each left table row which is unmatched, create a row placing NULL values in the right table columns. For each right table row which is unmatched, create a row placing NULL values in the left table columns.

Join Examples

Name	Age
Edna	63
Tim	37
Jim	17
Yolanda	25
Bernie	75
Tony	10

Name	Pet
Patrick	Rover
Edna	Fifi
Edna	Fido
Edna	Fluffy
Jim	Scruffy



Inner Join

Name	Age	Pet
Edna	63	Fifi
Edna	63	Fido
Edna	63	Fluffy
Jim	17	Scruffy

Left Join

Name	Age	Pet
Edna	63	Fifi
Edna	63	Fido
Edna	63	Fluffy
Tim	27	NULL
Jim	17	Scruffy
Yolanda	25	NULL
Bernie	75	NULL
Tony	10	NULL

Outer Join

Name	Age	Pet
Patrick	NULL	Rover
Edna	63	Fifi
Edna	63	Fido
Edna	63	Fluffy
Tim	27	NULL
Jim	17	Scruffy
Yolanda	25	NULL
Bernie	75	NULL
Tony	10	NULL

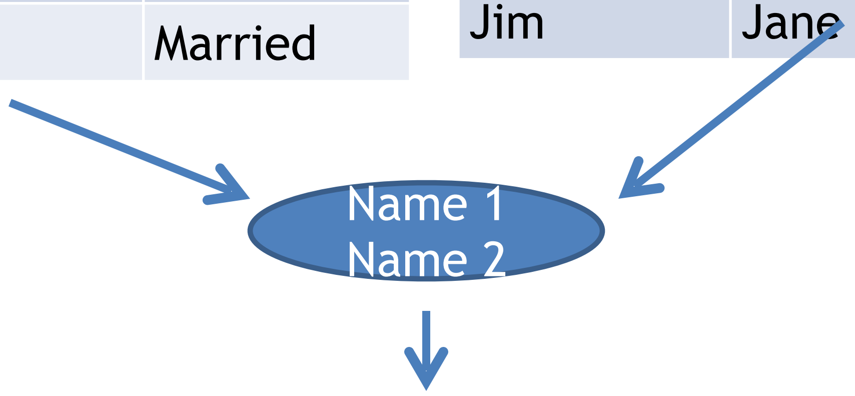
Multi Column Joins

- Joins can also be conducted on multiple columns.
- If n columns in Table A are compared to n columns in Table B, then a match occurs when all items across the compared columns match.

Multi Column Join (Inner)

Name 1	Name 2	Status
Jim	Jane	Friends
Jane	Jim	Married
Jane	Tony	Coworkers
Jim	John	Married

Name 1	Name 2	Connection
Jim	Jane	Pinterest
Jane	Tony	Facebook
Jim	Jane	Facebook



Name 1	Name 2	Status	Connection
Jim	Jane	Friends	Pinterest
Jane	Tony	Coworkers	Facebook
Jim	Jane	Friends	Facebook

Other Uses

- Joins for filtering
  - When data sets are large.
  - When the list of values to use in filtering is large.
- Semi Join
  - Return the left table when there is a right table match.
- Anti Join
  - Return the left table when there is no right table match.

Best Practices

- Before joining your data
  - Get table counts
  - Review data types
  - Look at the table headers (top 5 or so columns)
  - Picture the table post join
    - Row count
    - Columns
    - Data types
- Know the gotchas
  - Replication
  - Loss of observations

Scenario One

- Two tables are compared by zip code
  - Table count for 1 is 40,000
  - Table count for 2 is 40,000
- Join is conducted and error thrown due to data type mismatch
  - String
  - Integer
- Table two's zip code column is converted to string
- Join is conducted successfully
  - Table count is 36,000

Scenario Two

- Two table are compared by zip code
  - Table count for 1 is 40,000
  - Table count for 2 is 40,000
- Join is conducted successfully
  - Table count is 0

Scenario Three

- Two tables are compared one with person (“First Last”), height, the other with person(“First Last”), country.
- Table counts
  - Table 1: 10,000
  - Table 2: 8,500
- Join is conducted on person
- Final table count
  - Table 14,875

Let’s Join Some Stuff

- What is the syntax for joining to Data Frames in R?
- Remember we are working in the Tidy Universe.

Syntax for Joins

- new_df <- inner_join(df1, df2, by=“joincol”)
- new_df <- outer_join(df1, df2, by=“joincol”)
- new_df <- left_join(df1, df2, by=“joincol”)

Syntax for Multicolumn Comparison

- This also works for joining columns with differing names

- new_df <- inner_join(df1, df2, by=c(“xcol” = “acol”, “ycol” = “bcol”))