05 - Large Data

R Workshop
- Data Formatting and Reshaping -

Outline

- Why Databases? data formats
- basic SQL syntax
- connecting to a database
- joining data from different sources

R and Memory

- R by default loads objects into main memory
- Size of data is limited by a machine's main memory
  (typically small and expensive compared to size and prices of hard disks)
- Large data needs to stay outside of R, for an analysis only summary statistics are loaded
  Different approaches, e.g. `bigMem`, `ff`, `bigLM`, databases

Database

- Databases consist of large storage elements
  tools for fast retrieval of individual information and (simple) data aggregations

- Different Languages, among them SQL, Structured Query Language
- We'll look at data structures first, then the language

Less duplication

More consistency

Normal form of data

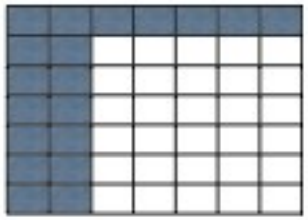
Small pieces joined together

Harder to edit

Harder to view

The key, ...

1st normal form



- Records in rows
- Variables in columns
- No duplicate rows
  (i.e. we could use row number as key)
- Order of rows and columns can not contain any information (i.e. we do not loose any information by reordering rows or columns)

Order of rows or columns

- If row order contains information, make it explicit: introduce another column variable 'Order' that contains the values 1 to number of rows
- If column order contains information, that is a sign that we store some additional information in headers. Introduce another variable that makes that explicit - usually that means that we have to re-format our table

Date	Weekly U.S. All Grades All Formulations Retail Gasoline Prices (Dollars per Gallon)	Weekly East Coast All Grades All Formulations Retail Gasoline Prices (Dollars per Gallon)	Weekly New England (PADD 1A) All Grades All Formulations Retail Gasoline Prices (Dollars	Weekly Central Atlantic (PADD 1B) All Grades All Formulations Retail Gasoline Prices (Dollars
4/5/93	1.068	1.04	1.065	1.068
4/12/93	1.079	1.047	1.073	1.072
4/19/93	1.079	1.054	1.074	1.077
4/26/93	1.086	1.059	1.076	1.08
5/3/93	1.086	1.062	1.08	1.084
5/10/93	1.097	1.069	1.091	1.09
5/17/93	1.106	1.073	1.[illegible]	1.095

Gas Prices Example

Date	Location	Weekly Price

..., the whole key ...

2nd normal form

- Violated when: Fact is about a subset of a key (when composite keys)
- To fix: create another dataset

Person	Date	Weight	Sex
James	1 Jan	205	Male
James	1 Feb	195	Male

Person	Date	Weight
James	1 Jan	205
James	1 Feb	195
James	1 Mar	190

Person	Sex
James	Male

… and nothing but the key

3rd normal form

- Violated when: Non-key field is a fact about another non-key field

Person	Zip Code	State
Heike Hofmann	50014	Iowa
John Smith	90210	California
...	...	...

Your Turn

- For the following table identify the key(s) and bring the table into 2nd Normal Form

Name	Major	ID	Date	Status
Never Ever	CS	1234	02-05	Absent
Never Ever	CS	1234	02-07	Absent
Equal Odds	Stats	5678	02-05	Present
Equal Odds	Stats	5678	02-07	Absent
Some Times	Math	4321	02-05	Present
Some Times	Math	4321	02-07	Absent
...	...	...	...	...

Normalized Tables

Student

ID	Name	Major
1234	Never Ever	CS
5678	Equal Odds	Stats
4321	Some Times	Math
...	...	...

Attendance

ID	Date	Status
1234	02-05	Absent
1234	02-07	Absent
5678	02-05	Present
5678	02-07	Absent
4321	02-05	Present
4321	02-07	Absent
...	...	...

SQL

- Structured Query Language (1970, E Codds)
- Programming language used for accessing data in a database
- ANSI standard since 1986, ISO standard since 1987
- Still some portability issues between software systems! Quite a few different SQL dialects.
- We'll mainly focus on SQL queries to access data

SELECT

- Selects data from the database

```
SELECT column_name(s)
FROM table_name
```

Student

ID	Name	Major
1234	Never Ever	Math
5678	Equal Odds	Stats
4321	Some Times	CS
...	...	...

Attendance

ID	Date	Status
1234	02-05	Absent
1234	02-07	Absent
5678	02-05	Present
5678	02-07	Absent
4321	02-05	Present
...	...	...

```
SELECT Name, Major
FROM Student
```

→

Name	Major
Never Ever	Math
Equal Odds	Stats
Some Times	CS
...	...

SELECT

Student

ID	Name	Major
1234	Never Ever	Math
5678	Equal Odds	Stats
4321	Some Times	CS
...	...	...

Attendance

ID	Date	Status
1234	02-05	Absent
1234	02-07	Absent
5678	02-05	Present
5678	02-07	Absent
4321	02-05	Present
4321	02-07	Absent
...	...	...

All

```
SELECT *
FROM Student
```

→

ID	Name	Major
1234	Never Ever	Math
5678	Equal Odds	Stats
4321	Some Times	CS
...	...	...

WHERE

Student

ID	Name	Major
1234	Never Ever	Math
5678	Equal Odds	Stats
4321	Some Times	CS
...	...	...

Attendance

ID	Date	Status
1234	02-05	Absent
1234	02-07	Absent
5678	02-05	Present
5678	02-07	Absent
4321	02-05	Present
4321	02-07	Absent
...	...	...

```
SELECT Name
FROM Student
WHERE Major='Math'
```

Name
Never Ever
...

Functions & Aggregates

Student

ID	Name	Major
1234	Never Ever	Math
5678	Equal Odds	Stats
4321	Some Times	CS
...	...	...

Attendance

ID	Date	Status
1234	02-05	Absent
1234	02-07	Absent
5678	02-05	Present
5678	02-07	Absent
4321	02-05	Present
4321	02-07	Absent
...	...	...

```
SELECT ID, count(ID)
FROM Attendance
WHERE Status='Absent'
GROUP BY ID
```

ID	Frequency
1234	2
5678	1
4321	1
...	

Functions

- COUNT
- AVG
- MAX
- MIN
- SUM
- ROUND
- LEN
- ...

http://www.w3schools.com/sql/sql_functions.asp

Your Turn

- Go to website http://www.w3schools.com/sql/sql_tryit.asp to try for yourself:
- What fields are in the table "customers"?
- Select the CompanyName and ContactName of customers that come from Germany
- Find a frequency breakdown of all customers by country.

Front-ends/Back-ends

- A front end is responsible for collecting input from the user and processing it to conform to the specification that back-end can execute.
- Need to connect to the database
- And execute queries

Accessing Databases

- Packages in R have Front-/Backend Set-up
- Back-end is the same for all database management systems (DBMS): done by DBI package
- Front-end depends on the DBMS, there is RMySQL, RSQLite, ROracle, ...

Packages DBI, RMySQL

- DBI is a general interface to DBMS
- RMySQL extends DBI for with specific functions to access mysql database
- You may need to install the mysql client in order to run RMySQL, if using own machine (http://biostat.mc.vanderbilt.edu/wiki/Main/RMySQL)
- From the thin client labs will need to ssh to `linux10.stat.iastate.edu`

DBI, RMySQL

- Link to Database:
  `dbDriver, dbConnect, dbDisconnect`
- Get Information:
  `dbListTables, dbListFields`
- Get Records:
  `dbReadTable, dbGetQuery, dbSendQuery`

Baseball data

- Full data set that plyr data set draws from http://www.baseball-databank.org
- Large collection of baseball statistics!

Connecting to the DB

```
> library(DBI)
> library(RMySQL)

> drv <- dbDriver("MySQL")
> co <- dbConnect(drv, user="2009Expo",
    password="R R0cks", port=3306,
    dbname="baseball", host=
    "headnode.stat.iastate.edu")

> dbListTables(co)
> dbListFields(co, "Batting")
```

Executing queries

```
> dbGetQuery(co, "SELECT count(*)
FROM Batting")
count(*)
1    92706
> df <- dbGetQuery(co, "SELECT * FROM
Batting")
> dim(df)
> head(df)
> ?dbGetQuery
```

Your Turn

- The table HallOfFame contains data on all baseball players inducted/considered for induction to the hall of fame.
- How many different players are in the table?
- What was the most recent year? First year?

Your Turn - Advanced

- The following information helps you connect to a large database of flight information
- Figure out, what information is available, how many flights there are in total, and whether Day of the week has an impact on flight delays
- ```
  co <- dbConnect(drv,
  user="2009Expo",
    password="R R0cks", port=3306,
    dbname="ontime", host=
    "headnode.stat.iastate.edu")
  ```
  ```