Querying and Filtering Data





Data Queries



A common task in data science

Querying: Accessing parts of the data based on given criteria

Basic and conditional queries with data.frame of R Base

SQL-like queries with data.table

Improved querying procedures on tibble format with dplyr

Simple character based queries



Running Queries with R Base



Querying Data Frames in R

Data querying is key in data science

- Dedicated language: SQL

R's querying features: Indexing, subsetting, filtering

Simple queries:

- Indexing operator []
- [row ID/'name', column ID/'name']

Define multiple values with concatenate:

- c(value1, value2)



Running Conditional Queries with R Base



Querying Data Frames in R

Querying with index positions []

Subsetting based on logical conditions

- 'Which observations exceed a given threshold?'
- 'Which observations match a given value?'

Logical operators: >, <, >=, <=, ==, &, |

Query function:

- subset(data.frame, test)

Selecting columns: select(column)



Logical Operators

Operator	Meaning
>	Greater than
<	Less than
==	Equal
>=	Greater than or equal
<=	Less than or equal



Logical Operators

Operator	Meaning
>	Greater than
<	Less than
==	Equal
>=	Greater than or equal
<=	Less than or equal
<u>!</u>	NOT



Logical Operators

Operator	Meaning
>	Greater than
<	Less than
==	Equal
>=	Greater than or equal
<=	Less than or equal
!	NOT
&	AND
	OR



Querying Data with Data.table



Advantages of Data.table Over R Base Queries



More intuitive, but standard methods work too

Requires less coding

Better performance even with big tables

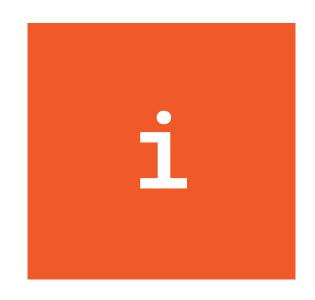
Further advantages: Ordered joins and great documentation

SQL like query commands:

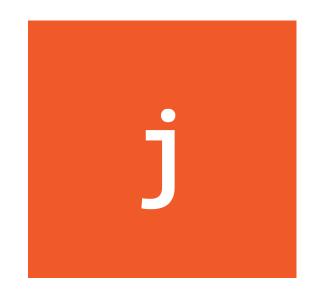
- data.table[i, j, by]



Data.table Query Syntax



Subset (rows) to be extracted based on a condition



Calculation to be performed on the subset



Grouping parameter that servers as a base for aggregation



Complex Queries with Data.table



Useful Functions for Queries

%between%

Subsetting by a range of values

%in%

Querying by value matching

order

Row ordering by reference column



Benefits of Data.table



Detailed documentation



Regular updates



Intuitive functionality



Tibble Manipulations with Library Dplyr



Tabular Data Structures in R

Class data.frame (R Base)

Class data.table (data.table)

Class tibble (dplyr or tibble)



Library Dplyr

Class tibble with backup class data.frame
Simple and consistent syntax
Great documentation with example code
Functions for table manipulation are called verbs



verb(data, criteria)

The General Syntax

The dataset can be a tibble or a data.frame



Single Table Verbs (Dplyr)

Reorder: arrange()

Pick observations:
 filter()

Pick variables:
 select()

Add new variable: mutate()

Collapse table:
 summarise()



verb(data, criteria)

The General Syntax

The dataset can be a tibble or a data.frame



Filtering and Reordering Tables

Filter

Select rows based on criteria

Slice

Select rows based on row ID

Arrange

Order the table based on a variable



Single Table Verbs with Dplyr



Single Table Verbs (Dplyr)

Reorder: arrange()

Pick observations:
 filter()

Pick variables:
 select()

Add new variable: mutate()

Collapse table:
 summarise()



Use the combination of distinct and select to explore the unique values of a variable



Single Table Verbs (Dplyr)

Reorder: arrange()

Pick observations:
 filter()

Pick variables:
 select()

Add new variable: mutate()

Collapse table:
 summarise()



Running Queries Based on Strings



Querying Based on Data Types





String Manipulations

Function gsub with regular expressions

Function str_detect from library stringr



Data Queries



Queries on data.frame

- Presented techniques work with improved data structures as well
- Indexing operator: []
- Function subset() for introducing logical conditions

Queries with data.table (i, j, by)

- Aggregations are available

Querying and manipulating tibbles and data.frames with dplyr's single table verbs

Library stringr for character based queries

