O'REILLY®

R-Powered Excel

Month/Year
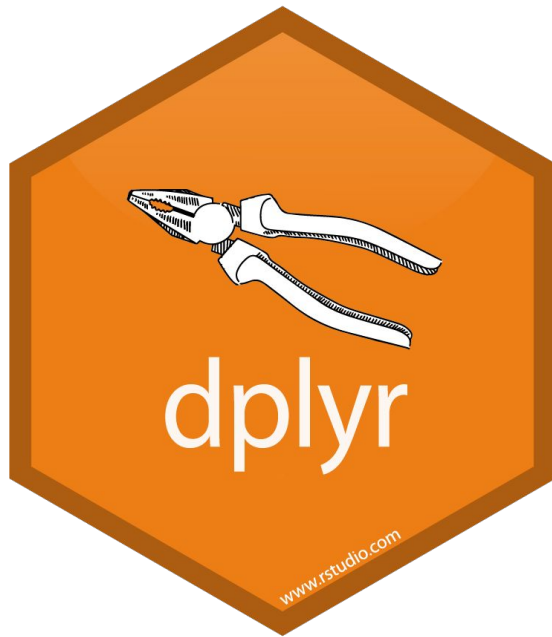
From PivotTables to dplyr manipulation

Demo: `pivot-dplyr-demo.xlsx`

`dplyr`: a grammar of data manipulation

Important `dplyr` "verbs"

Function	What it does
`mutate()`	Creates new columns based on exsiting columns
`select()`	Selects selected columns
`rename()`	Renames selected columns
`arrange()`	Reorders rows
`filter()`	Selects rows based on condition
`group_by()`	Groups records by selected columns
`summarise()`	Aggregates values for each group

Demo: `dplyr-manipulation-demo.r`

Drills: `dplyr` manipulation

1. Create a new data frame, `iris_log`, based on the existing `iris` data frame. Continue assigning the output to this object.
2. Create `Petal.Length_log` and `Petal.Width_log` which are log-transformation of their respective fields.
3. Filter this data frame to contain only records from the `setosa` species.
4. Sub-set this data frame to keep only the `Species`, `Petal.Length_log` and `Petal.Width_log` fields.

1. Create a new data frame, `iris_log`, based on the existing `iris` data frame. Continue assigning the output to this object.
2. Create `Petal.Length_log` and `Petal.Width_log` which are log-transformation of their respective fields.
3. Filter this data frame to contain only records from the `setosa` species.
4. Sub-set this data frame to keep only the `Species`, `Petal.Length_log` and `Petal.Width_log` fields.

Solutions:
`dplyr-manipulation-drills.r`

Drills: `dplyr` manipulation

dplyr: the power of the pipe

Important `dplyr` "verbs"

Function	What it does
`mutate()`	Creates new columns based on exsiting columns
`select()`	Selects selected columns
`rename()`	Renames selected columns
`arrange()`	Reorders rows
`filter()`	Selects rows based on condition
`group_by()`	Groups records by selected columns
`summarise()`	Aggregates values for each group
`%>%`	**Connect multiple verbs into a "pipeline"**

Demo: `dplyr-pipe-demo.r`

Using the `teams` data frame from Lahman:

1. Get average annual attendance for all NL teams, sorted from high to low.

Drills: `dplyr` pipeline

Using the `teams` data frame from Lahman:

1. Get average annual attendance for all NL teams, sorted from high to low.

Drills: `dplyr` pipeline

From `VLOOKUP()` to dplyr `join()`

Demo: `lookup-join-demo.xlsx`

From `VLOOKUP()` duct tape to `join()` welder

Drills: Joining data frames

Using the `Managers` and `AwardsManagers` data frames from `Lahman` (ignore the warnings):

1. Return the join of records found in both tables. Keep all fields.
2. Return the join of records found in both tables. Keep all fields *except* `Managers$rank`.
3. Return the join of records for all found in `Managers`.
   a. How many more rows does this have than the results of 1?

Using the `Managers` and `AwardsManagers` data frames from `Lahman` (ignore the warnings):

1. Return the join of records found in both tables. Keep all fields.
2. Return the join of records found in both tables. Keep all fields *except* `Managers$rank`.
3. Return the join of records for all found in `Managers`.
   a. How many more rows does this have than the results of 1?

Solutions: `join-drills.r`

Drills: Joining data frames

Demo: `join-demo.r`

O'REILLY®

Questions?