Manipulating data with dplyr

Based on R-ecology lesson - Data carpentry

Marco Chiapello, PhD

March 29, 2017

What is dplyr?

- **dplyr** is a package for making data manipulation easier
  - Packages in R are basically sets of additional functions that let you do more stuff

What is dplyr?

- **dplyr** is a package for making data manipulation easier
  - Packages in R are basically sets of additional functions that let you do more stuff
- dplyr provides **easy tools** for the most common **data manipulation** tasks

What is dplyr?

- **dplyr** is a package for making data manipulation easier
  - Packages in R are basically sets of additional functions that let you do more stuff
- dplyr provides **easy tools** for the most common **data manipulation** tasks
- dplyr addresses this by porting much of the computation to C++

What is dplyr?

- **dplyr** is a package for making data manipulation easier
  - Packages in R are basically sets of additional functions that let you do more stuff
- dplyr provides **easy tools** for the most common **data manipulation** tasks
- dplyr addresses this by porting much of the computation to C++
- An additional feature is the ability to work directly with data stored in an **external database**

What is dplyr?

Before start to dig into dplyr functions we learn how to import data into R

- To download the data, run the following:

```
download.file("https://ndownloader.figshare.com/files/2292169",
              "portal_data_joined.csv")
```

- You are now ready to load the data:

```
surveys <- read.csv('portal_data_joined.csv')
```

What is dplyr?

Before start to dig into dplyr functions we learn how to import data into R

- To download the data, run the following:

```
download.file("https://ndownloader.figshare.com/files/2292169",
              "portal_data_joined.csv")
```

- You are now ready to load the data:

```
surveys <- read.csv('portal_data_joined.csv')
```

- Converts data to tbl class. tbl's are easier to examine than data frames. R displays only the data that fits onscreen

```
surveys <- tbl_df(surveys)
```

What is dplyr?

We're going to learn some of the *most common dplyr functions:*

- **select**
- **filter**
- **arrange**
- **mutate**
- **group_by**
- **summarize**

Select

- To select columns of a data frame, use select()
  - The **first argument** to this function is the **data frame**
  - The **subsequent arguments** are the **columns to keep**

```
select(surveys, plot_id, species_id, weight)
```

Select

- To select columns of a data frame, use select()
  - The **first argument** to this function is the **data frame**
  - The **subsequent arguments** are the **columns to keep**

```
select(surveys, plot_id, species_id, weight)
```

- select is much more powerful than just select the interest columns
  - You can remove one column

```
select(surveys, -weight)
```

Select

- To select columns of a data frame, use select()
  - The **first argument** to this function is the **data frame**
  - The **subsequent arguments** are the **columns to keep**

```
select(surveys, plot_id, species_id, weight)
```

- select is much more powerful than just select the interest columns
  - You can remove one column

```
select(surveys, -weight)
```

```
*  Select columns whose name contains a character string
```

```
select(surveys, contains("ec"))
```

Select

- Select columns whose name starts with a character string

```
select(surveys, starts_with("s"))
```

Select

- Select columns whose name starts with a character string

```
select(surveys, starts_with("s"))
```

- Select all columns between Sepal.Length and Petal.Width (inclusive).

```
select(surveys, plot_id:weight)
```

Select

- Select columns whose name starts with a character string

```
select(surveys, starts_with("s"))
```

- Select all columns between Sepal.Length and Petal.Width (inclusive).

```
select(surveys, plot_id:weight)
```

- Select every column

```
select(surveys, weight, everything())
```