Data Manipulation 1: Exercises

This document accompanies the Data Manipulation Part 1 tutorial: http://rpubs.com/NateByers/DataManip1. These exercises use data frames from the region5air library. Run the following code to clean out your global environment and load the data you need:

rm(list = ls())  
library(dplyr)  
library(region5air)  
data(airdata)  
data(chicago\_air)

## Exercises

1. Use select() on the airdata data frame to create a monitors data frame with columns "site", "lat", "lon", and "GISDatum".

[Solution 1](#ex1)

1. Use arrange() on airdata to order it by site then by parameter then by datetime.

[Solution 2](#ex2)

1. Use filter() on airdata to create a pm data frame of PM2.5 measurements (AQS code 88101) from site 840180890022 with hourly values above 35 ug/m3. **Hint**: The "site" column is a character class and the "parameter"" and "value" columns have a numeric class. Use quotes around characters and unquoted numbers for numeric values.

[Solution 3](#ex3)

#### Advanced Exercises

1. From chicago\_air, create a data frame with readings between September 1 and September 30 where temperature values were at or above 90 degrees Fahrenheit.

[Solution 4](#ex4)

1. Use filter() and %in% to filter the airdata data frame down to just ozone (44201) and PM2.5 (88101). Remember, the "parameter" column is a character class, so use quotes around the AQS parameter codes.

[Solution 5](#ex5)

## Solutions

#### Solution 1

monitors <- select(airdata, site, lat, lon, GISDatum)

This returns a very long data frame with many duplicate values. You can use the distinct() function from dplyr to remove the duplicated rows.

# look at the dimensions of the data frame  
# the first number is the total number of rows, the second is the columns  
dim(monitors)

## [1] 367595 4

# remove duplicates   
monitors <- distinct(monitors)  
  
dim(monitors)

## [1] 26 4

[Back to exercises](#ex)

#### Solution 2

airdata <- arrange(airdata, site, parameter, datetime)

[Back to exercises](#ex)

#### Solution 3

pm <- filter(airdata, parameter == 88101, site == "840181270024", value > 35)

[Back to exercises](#ex)

#### Solution 4

If we want to filter using date ranges, we need to make sure that date values are one of the date classes. In the chicago\_air the date column is the character class, not a date class.

class(chicago\_air$date)

## [1] "character"

We can covert it to the Date class.

# no need to supply a format paramter--the date column is already in default format  
chicago\_air$date <- as.Date(chicago\_air$date)  
class(chicago\_air$date)

## [1] "Date"

Now we can filter using dates.

filter(chicago\_air, date >= as.Date("2013-09-01"), date <= as.Date("2013-09-30"),  
 temp >= 90)

## date ozone temp solar month weekday  
## 1 2013-09-10 0.059 91 1.15 9 3

Or we can simply filter on the month of September,

filter(chicago\_air, month == 9, temp >= 90)

## date ozone temp solar month weekday  
## 1 2013-09-10 0.059 91 1.15 9 3

[Back to exercises](#ex)

#### Solution 5

oz\_pm <- filter(airdata, parameter %in% c("44201", "88101"))

[Back to exercises](#ex)