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:

```r
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

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

Solution 2

3. Use `filter()` on `airdata` to create a `pm` data frame of $PM_{2.5}$ measurements (AQS code 88101) from site 840180890022 with hourly values above 35 ug/m^3. **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

Advanced Exercises

4. 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

5. Use `filter()` and `%in%` to filter the `airdata` data frame down to just ozone (44201) and $PM_{2.5}$ (88101). Remember, the "parameter" column is a character class, so use quotes around the AQS parameter codes.

Solution 5

Solutions

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

Solution 1 This returns a very long data frame with many duplicate values. You can use the `distinct()` function from `dplyr` to remove the duplicated rows.

```r
# 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
```

```r
# remove duplicates
monitors <- distinct(monitors)

dim(monitors)
```

```
## [1] 26  4
```

Back to exercises

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

Solution 2 Back to exercises

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

Solution 3 Back to exercises

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.

```r
class(chicago_air$date)
```

```
## [1] "character"
```

We can covert it to the Date class.

```r
# 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.

```r
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,

```r
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

```r
oz_pm <- filter(airdata, parameter %in% c("44201", "88101"))
```

Solution 5 Back to exercises