

# Introduction to dplyr: Exercises

*Northwestern University Computational Research Day*

*18 April 2017*

These exercises will use datasets from the `nycflights13` library. In order to load the dataset we will need to first install the library (if it is not already installed), load the library, and then attach the dataset.

```
install.packages("nycflights13")
library(nycflights13)
```

## Single Table Verbs

1. Create two datasets called `flights_am` and `flights_pm` that contain flights that are scheduled to leave before 1200 and after 1200.
2. For both `flights_am` and `flights_pm`, do the following. Save the resulting datasets as `fl_am_full` and `fl_pm_full`.
  - (a) Add a column that indicates if a flight departed on time. Call it `dep_ontime`. Define 'on time' as within 10 minutes of scheduled departure.
  - (b) Add a column that indicates if a flight arrived on time. Call it `arr_ontime`. Define 'on time' as within 10 minutes of scheduled arrival.
  - (c) Add a column that indicates if a flight left more than 10 minutes late, but arrived on time. Call it `latedep_ontimearr`.
  - (d) Add a column that reports the date (year-month-day) of the flight. Call it `date`. (Hint: you may need the `as.Date()` and `paste()` functions, along with the `year`, `month`, and `day` fields.)
3. Create two new datasets `fl_am` and `fl_pm` by selecting the following columns from `flights_am_full` and `flights_pm_full`:
  - (a) Drop `year`, `month`, and `day` (since you have all of that in `date`).
  - (b) Rename scheduled departure time as `sched_dep`
  - (c) Rename scheduled arrival time as `sched_arr`
4. Take both `fl_am` and `fl_pm` and create some summary statistics.
  - (a) Proportion of flights that depart and arrive on time.
  - (b) Proportion of flights that depart late but arrive on time.
  - (c) Average air time
  - (d) Average and standard deviation of flight distances

## Combining Single-Table Verbs

5. Above we used a series of single-table verbs to create `fl_am` and `fl_pm`. Re-do this using only a single series of pipes commands. For both `fl_am` and `fl_pm`:
  - (a) Take the `month`, `day`, and `year` columns and create `date`. Then drop `month`, `day`, and `year`.
  - (b) Rename `sched_arr_time` and `sched_dep_time` to `sched_arr` and `sched_dep`.
  - (c) Adds columns to indicate if flights arrive on time, leave on time, and arrive on time after leaving late.
6. In the `weather` dataset, use the `month`, `day`, and `year` columns and create `date` column as above. Then drop the `month`, `day`, and `year` columns. Save the results in `wthr`. Do all of this using one set of piped commands.
7. In the `planes` dataset, determine:
  - (a) What engines go on the largest/smallest planes (by number of seats).
  - (b) What engines go on the fastest/slowest planes (on average).

## Grouping

8. Using the `flights` data set, determine:
  - (a) The number of flights each plane (`tailnum`) flies, as well as the flights it flies from each airport in the dataset.
  - (b) The average air time (`air_time`) for each plane.
  - (c) The average departure and arrival delays for each carrier.
9. Using the weather data, determine which weather station (`origin`) had:
  - (a) The fastest wind gust (`wind_gust`).
  - (b) The highest average wind speed (`wind_speed`).
  - (c) The most variable (in standard deviation) wind speeds.
  - (d) The worst average visibility (`visib`).

(Hint: try using one set of piped commands to get all of the data you might need in a-d, and store it as `wth`. Then you just need to use the `arrange()` function on `wth`.)
10. Using the planes data, determine:
  - (a) Which manufacturers have the most/fewest planes in the data.
  - (b) Which manufacturer has the oldest/newest (on average) planes in the data. (Hint: create a column called `age` and set `age = 2017 - year`)
  - (c) Which year saw the most/least amount of planes made.
  - (d) What type of engine is most/least common.

## Joins

11. Which airlines had some of the windiest flights? (Hint: join `flights` and `weather` and group by `carrier`).
12. Which airline(s) uses the oldest plane in the data? (Hint: join `flights` and `planes` and use `summarize`)
13. Which airline(s) has the oldest mean or median plane in the data? (Hint: take only distinct `tailnum-carrier` pairs from `flights`, join that with `planes`, mutate to include `age`, and then `summarize`).
14. Obtain a list of planes that flew to destinations above  $45^{\circ}$  latitude. What is the newest plane? What is the average size of these planes? How many of these flights emanated from each of the origin airports in the data? (Hint: join `flights` and `airports`, filter by `lat`, then join with `planes`)

## Putting it All Together

Create one dataset that contains all of the information in the `flights`, `airports`, `weather`, and `planes` datasets. Make sure to replace the day, month, and year columns with one column called 'date'. Use this dataset to answer questions 7-12 above.