# DCF Week 3 Activity

*Data and Computing Fundamentals*

The `OrdwayBirds` data table is a historical record of birds captured and released at the Katharine Ordway Natural History Study Area, a 278-acre preserve in Inver Grove Heights, Minnesota, owned and managed by Macalester College. Originally written by hand in a field notebook, the entries have been transcribed into electronic format under the supervision of Jerald Dosch, the Dept. of Biology, Macalester College.

The data are "dirty." This refers to the existence of possibly spurious or incorrect entries, often caused by data-entry errors.

One of the flaws in the data table is in the `SpeciesName` variable. This is intended to identify the species of each of the birds. Often, though, the spelling varies among birds of the same biological species. This leads to mis-classification of birds.

Fortunately, the errors are easy to correct. The data table `OrdwaySpeciesNames` collects together all the variant spellings. Entry by entry, each mis-spelling was translated into a standardized spelling. Thus, `join()` can be used to correct the mis-spellings in the `OrdwayBirds` table.

You are going to look at the month-to-month presence of different species. Think of your assignment as creating a manual for birders to guide them to the correct time of year to visit Ordway to see a particular species.

### In pairs, using the computer

Construct and run on the computer the commands to carry out the following tasks. To do this, open up an Rmd document from the DCF template, saving it under the name `Birds-XXX.Rmd` (where XXX is your initials or other identifier unique to you). Making sure to change the title and author to something appropriate, and to change the quoted character string on line 12 to match the name of your Rmd file.

To keep things simple for later on, cut and paste this command into a chunk at the start of the document.

```
OrdwayBirds <-
  OrdwayBirds %>%
  select( SpeciesName, Month, Day ) %>%
  mutate( Month=as.numeric(Month), Day=as.numeric(Day))
```

After that chunk, add text to describe briefly what each of the first three lines in the statement does.

- Line 1 does ...
- Line 2 does ...
- Line 3 does ...

Line 4 is part of the data cleaning process. Do to data-entry errors, the `Month` and `Day` were stored as categorical variables. They make more sense as numerical variables. Line 4 does the conversion.

**Step 1.** Including mis-spellings, how many different species are their in the `OrdwayBirds` data?

Compare this to the number of different species in the `SpeciesNameCleaned` variable in `OrdwaySpeciesNames`.

In `OrdwayBirds`, the case is the capture of an individual bird. Imagine the data table that contains the answer to the above question. Will it have the same meaning to a case? Knowing this will help you choose the appropriate data verb:

- Output table has the same case meaning as the input: `mutate()`, `arrange()`, `filter()`, `group_by()`, `join()`, `select()`
- Output table has a different case meaning than the input: `summarise()`

You will find it helpful, as an argument to to the data verb you choose, to know about the `n_distinct()` function, which counts the number of unique values in a variable.

**Step 2** Use the `OrdwaySpeciesNames` table to create a new data table that corrects the mis-spellings in `SpeciesNames`. This is simple matter of using `inner_join()`. Before actually carrying out the join, look at the names of the variables in `OrdwaySpeciesNames` and `OrdwayBirds`.

- Which variable(s) will be used for matching cases.
- What will be the variable(s) that will be added .

**Step 3** Count how many bird captures there are of each of the (corrected) species? You can call the variable that contains the count `count`. Arrange this into descending order from the species with the most birds, and look through the list. Hint: Remember `n()`. Also, one of the arguments to one of the data verbs will be `desc( count )` to arrange the cases into descending order.

Define for yourself a "major species" as a species with more than a particular threshold count. Set your threshold so that there are 5 or 6 species designated a major.

Filter to produce a data table with only the birds that belong to a major species. (Hint: Remember that summary functions can be used case-by-case when filtering or mutating a data table that has been grouped.) Save the output in a table called `Majors`.

**Step 4** When you have correctly produced `Majors`, write a command that produces the month-by-month count of each of the major species. Call this table `ByMonth`.

Display this month-by-month count with a bar chart arranged in a way that you think tells the story of what time of year the various species appear. You can use `mBar()` to explore different possibilities. Due to a bad design choice in `mBar()`, you need to change the `Month` variable back to categorical. These statements will do the job:

```
ForMBar <-
  ByMonth %>%
  mutate( Month=as.factor(Month) )
mBar( ForMBar)
```

You can use the "Show Expression" button in `mBar()` to create an expression that you can cut and paste into a chunk in your Rmd document, so that the graph gets created when you compile it. (`mBar` should not be a statement in your Rmd file: it needs to be used interactively from the console.)

Once you have the graph, use it to answer these questions:

- which specie(s) are present year-round?
- which specie(s) are migratory, that is, primarily present in one or two seasons?

### Joining with your group

Compare the answers the different members of your group came up with.

Next, construct on the whiteboard the commands to answer each of these questions:

1. What is the peak month for each major species?

2. Which major species that are seen in good numbers for at least 6 months of the year? (`n_distinct()` may be useful. Also, you can use as part of the comparison expression `>=6`.)

In constructing the appropriate commands, consider these questions:

- What data table should be the input: `Majors` or `ByMonth`?
- Which data verb to use? Hint: `group_by()`, `filter()` and `summarise()` may be useful.

Once you and your group have written down commands that you think will work, implement them, checking the outputs against the bar chart you made.

**Extra:**

- Identify, for the major species over each month, whether they tend to be seen almost only at the beginning or end of the month.