Selecting Columns in a DataFrame

Author: Cole Brookson **Date:** 20 July 2022

Often it is useful to be able to select only a select number of columns in a dataframe, in conjunction or separately from filtering the rows of a dataframe. We can do this to simply re-size the dataframe to only what we need for a particular future analysis, or to isolate the part of a dataframe we need to perform other operations on.

Selecting Columns

To do this, we can select a particular set of columns. Let's use the ice dataframe example from the `lterdatasampler` package:

```
# load packages
library(lterdatasampler)
library(tidyverse)

# load in dataframe we'll work with
ice <- lterdatasampler::ntl_icecover
```

We can look at the columns in this dataframe:

```
names(ice)
```

```
## [1] "lakeid"       "ice_on"       "ice_off"      "ice_duration" "year"
```

We may want to make a smaller dataframe with only the columns `year` and `ice_duration`. We can do this with the `dplyr` function `select()`:

```
small_ice <- ice %>%
  dplyr::select(year, ice_duration)
names(small_ice)
```

```
## [1] "year"         "ice_duration"
```

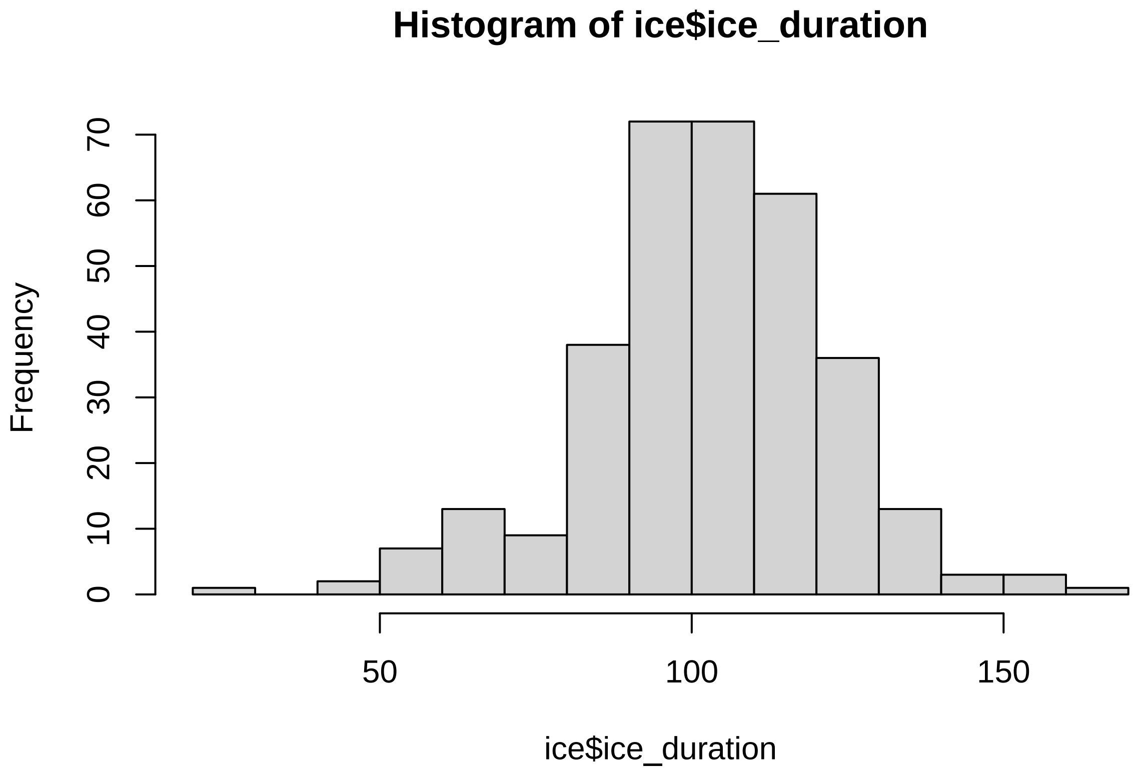
And we can see this works as expected.

Selecting & Filtering

It is always useful to be able to perform multiple operations on a dataframe at once, since it's rarely the case that we only need to do a single thing before our data object is ready for analysis.

We can "pipe" our operations together using the `dplyr` pipes. For example, we could select the same columns as we did already, but also filter `ice_duration` at the same time. Let's see what values that variable can take on:

```
# hist() will just give us a histogram of the values
hist(ice$ice_duration)
```



So we might want to filter our data to only values below 100. We can do that like this:

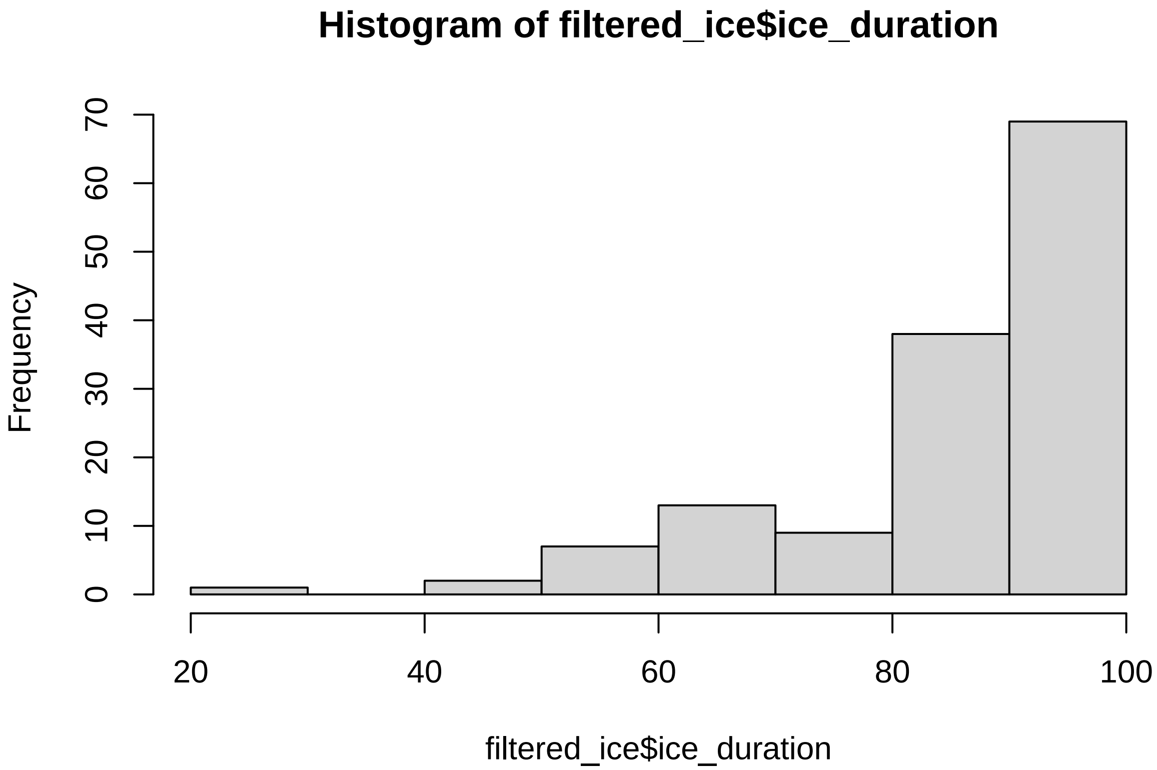
```
filtered_ice <- ice %>%
  dplyr::filter(ice_duration < 100) %>%
  dplyr::select(year, ice_duration)
```

Now we can look at the columns we have in this new object and also what values `ice_duration` takes on to check it worked correctly:

```
names(filtered_ice)
```

```
## [1] "year"         "ice_duration"
```

```
hist(filtered_ice$ice_duration)
```



We can see this worked as expected!