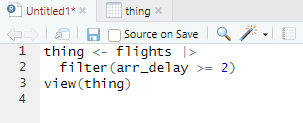
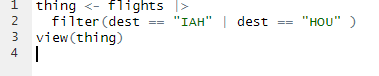
3.2.5 Exercises

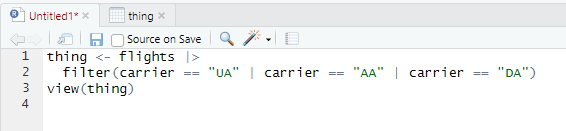
1. In a single pipeline for each condition, find all flights that meet the condition:
   * Had an arrival delay of two or more hours



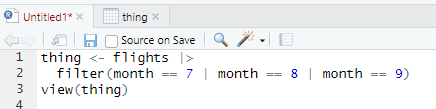
* + Flew to Houston (IAH or HOU)



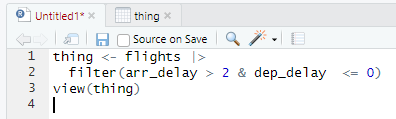
* + Were operated by United, American, or Delta



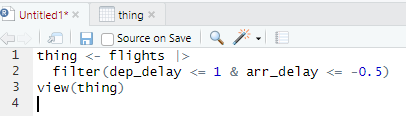
* + Departed in summer (July, August, and September)



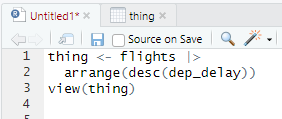
* + Arrived more than two hours late, but didn’t leave late

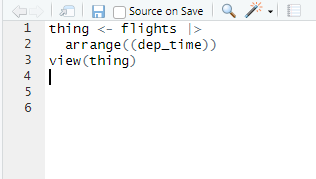


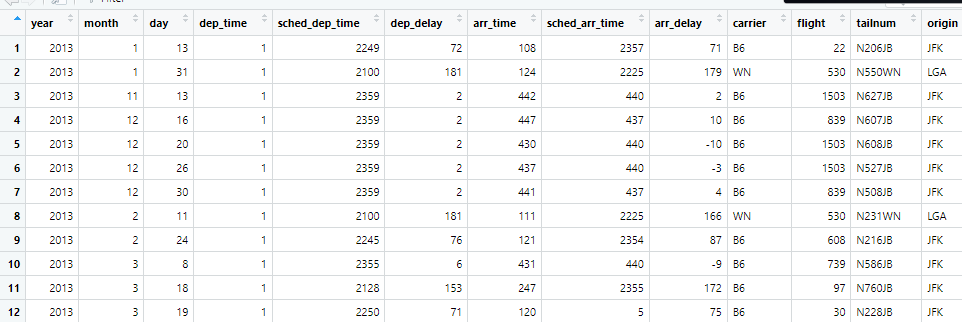
* + Were delayed by at least an hour, but made up over 30 minutes in flight



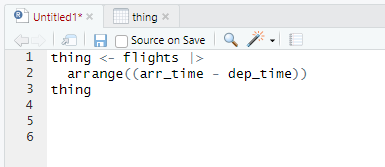
2. Sort flights to find the flights with longest departure delays. Find the flights that left earliest in the morning.

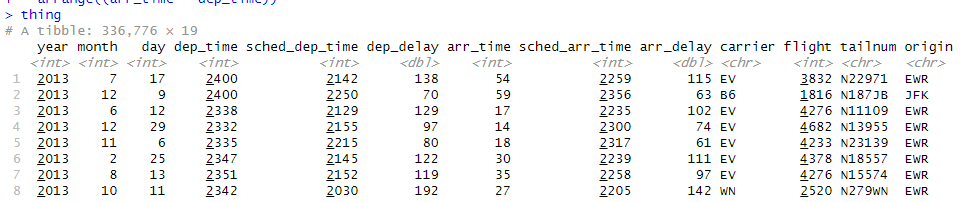






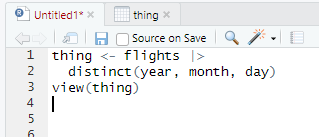
3. Sort flights to find the fastest flights. (Hint: Try including a math calculation inside of your function.)





4. Was there a flight on every day of 2013?

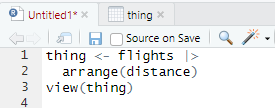
Yes there was a a flight on every day of 2013





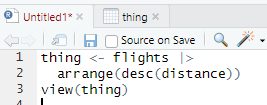
5. Which flights traveled the farthest distance? Which traveled the least distance?

Shortest Distance: 17 miles





Longest Distance: 4983 miles





6. Does it matter what order you used [filter()](https://dplyr.tidyverse.org/reference/filter.html) and [arrange()](https://dplyr.tidyverse.org/reference/arrange.html) if you’re using both? Why/why not? Think about the results and how much work the functions would have to do.

It does matter which order you use filter() and arrange() when using both. Using filter() and then arrange() will arrange the results of the filter, while using arrange() before filter() will arrange the values of the table before filtering, including filtered out values into the arrange order.

3.3.5 Exercises

1. Compare dep\_time, sched\_dep\_time, and dep\_delay. How would you expect those three numbers to be related?

If dep\_time is less than sched\_dep\_time, dep\_delay will be less. If dep\_time is greater than sched\_dep\_time, dep\_delay will be greater.

2. Brainstorm as many ways as possible to select dep\_time, dep\_delay, arr\_time, and arr\_delay from flights.

* select(dep\_time, dep\_delay, arr\_time, arr\_delay)
* select(dep\_time, dep\_delay:arr\_time, arr\_delay)
* select(dep\_time:arr\_delay - contains(“sched”))

3. What happens if you specify the name of the same variable multiple times in a [select()](https://dplyr.tidyverse.org/reference/select.html) call?

It only shows the variable once.

4. What does the [any\_of()](https://tidyselect.r-lib.org/reference/all_of.html) function do? Why might it be helpful in conjunction with this vector?

variables <- [c](https://rdrr.io/r/base/c.html)("year", "month", "day", "dep\_delay", "arr\_delay")

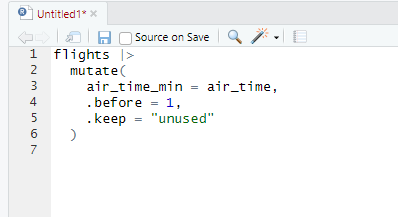
The any\_of() function calls all of the possible given variables and does not throw an error if any of the variables are empty or missing. This could be helpful in defining the variables vector, as it would pass through variables that might have a missing value in the row instead of getting stuck on an error.

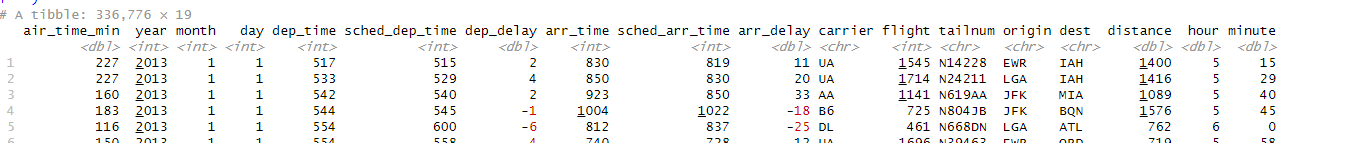
5. Does the result of running the following code surprise you? How do the select helpers deal with upper and lower case by default? How can you change that default?

flights |> [select](https://dplyr.tidyverse.org/reference/select.html)([contains](https://tidyselect.r-lib.org/reference/starts_with.html)("TIME"))

The result did surprise me. I expected the contains() method to be case sensitive. In reality, the method was not case sensitive. It is possible to change this by changing ignore.case = FALSE.

6. Rename air\_time to air\_time\_min to indicate units of measurement and move it to the beginning of the data frame.



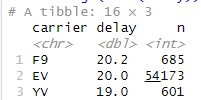
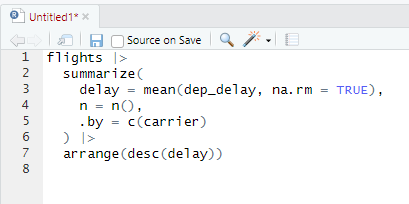


7. Why doesn’t the following work, and what does the error mean?

By selecting the variable tailnum, a new dataframe is created that only contains the aforementioned column. This new dataframe was then piped into an arrange statement for a column that no longer existed. The error itself means it cannot find the variable arr\_delay in the dataframe.

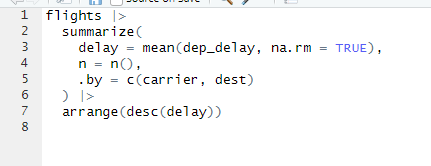
3.5.7 Exercises

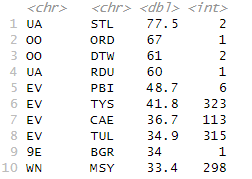
1. Which carrier has the worst average delays? Challenge: can you disentangle the effects of bad airports vs. bad carriers? Why/why not? (Hint: think about flights |> group\_by(carrier, dest) |> summarize(n()))



The F9 carrier has the worst delays with an average of 20.2.

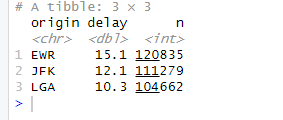
And it is somewhat possible to dientable the effects of bad airports vs. bad carriers.





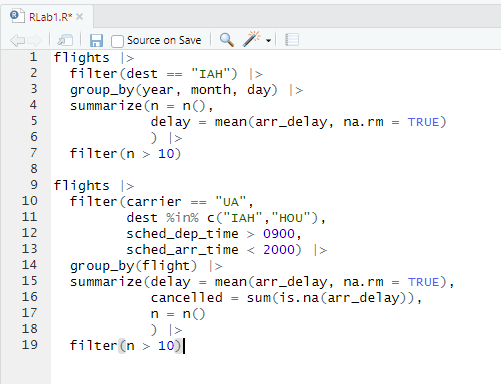
Here we can see that the OO and EV carriers have terrible delays regardless of what airport they are launched from.

2. Find the flights that are most delayed upon departure from each destination.



4.6 Exercises

1. Restyle the following pipelines following the guidelines above.



5.2.1 Exercises

1. For each of the sample tables, describe what each observation and each column represents.

For each table the country column holds observations containing the name of each country, and the column holds observations containing the year in which the data was found. For table 1, the cases column holds observations for the number of cases of tuberculosis in each country during that respective year. Similarly, the population column for table 1 contains the population of each country during the labeled year. Table 2 adds a type column, which is either labeled population or case. The following count column then gives the number with the associated type. Table 3 collapses the cases and population numbers from table 1 into one fraction. These fractions are held in a column named rate.

1. Sketch out the process you’d use to calculate the rate for table2 and table3. You will need to perform four operations:
   1. Extract the number of TB cases per country per year.
   2. Extract the matching population per country per year.
   3. Divide cases by population, and multiply by 10000.
   4. Store back in the appropriate place.

Table 2:

* Select row where type = case
* Save count for selected row
* Save year and country for selected row
* Filter for saved country and year
* Select row where type = population
* Save count for selected row
* Divide first saved count by second saved count and multiply by 10000
* Mutate into new rate column
* Repeat for every country and year

Table 3:

* Select rate from row
* Save year and country
* Save number before / as separate string
* Save number after / as separate string
* Convert strings to integers
* Divide number before / by number after / and multiply by 10000
* Mutate new rate column using new number and match with saved year and country
* Repeat