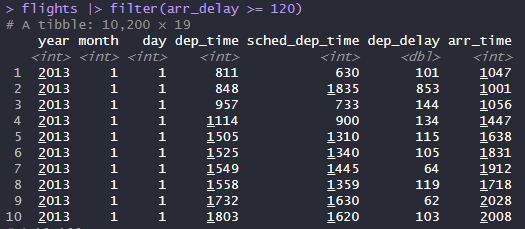
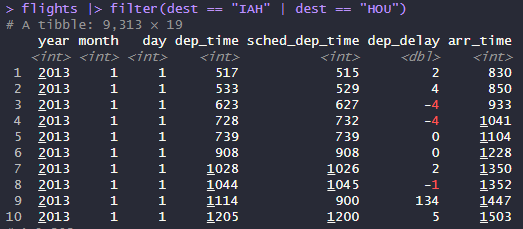
3.2.5 Exercises

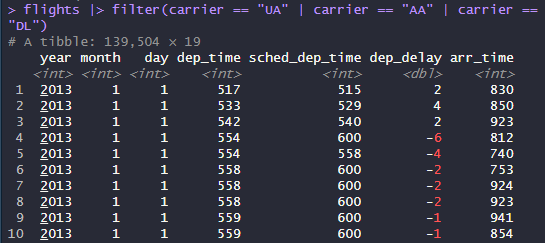
1. In a single pipeline for each condition, find all flights that meet the condition:
   * Had an arrival delay of two or more hours



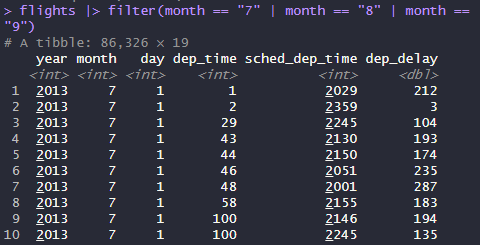
* + Flew to Houston (IAH or HOU)



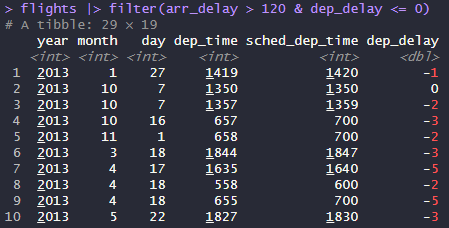
* + Were operated by United, American, or Delta



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



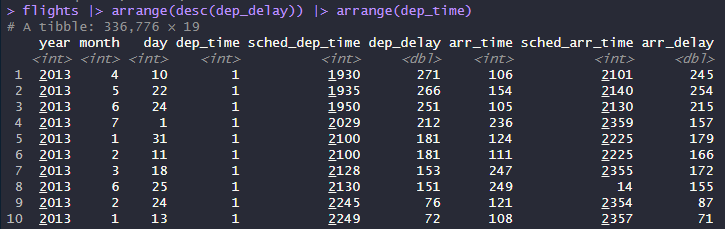
* + Arrived more than two hours late, but didn’t leave late



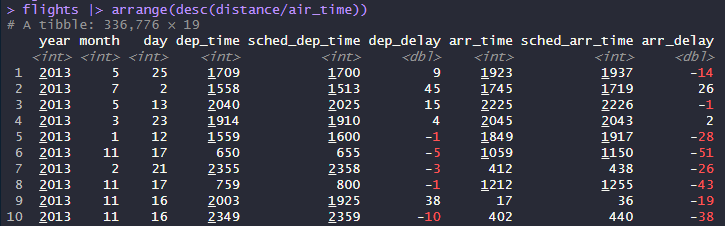
* + Were delayed by at least an hour, but made up over 30 minutes in flight



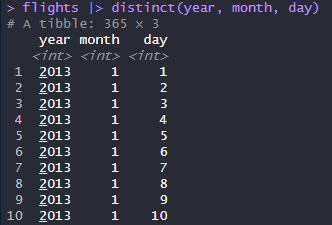
1. Sort flights to find the flights with longest departure delays. Find the flights that left earliest in the morning.



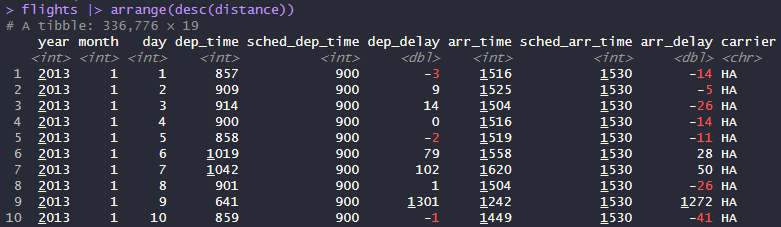
1. Sort flights to find the fastest flights. (Hint: Try including a math calculation inside of your function.)

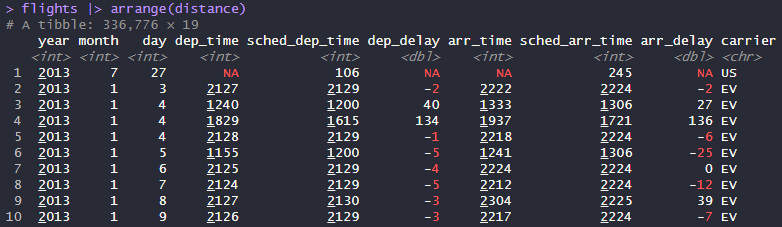


1. Was there a flight on every day of 2013?



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



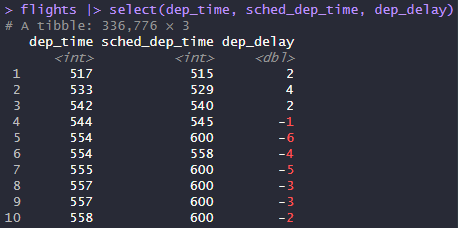


1. Does it matter what order you used [filter()](https://dplyr.tidyverse.org/reference/filter.html) and [arrange()](https://dplyr.tidyverse.org/reference/arrange.html) if you’re using both? Why/why not? Think about the results and how much work the functions would have to do.

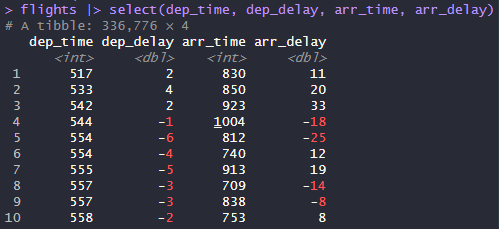
It does not matter the order if you use filter() and arrange() because the results would be the same. However, using filter() then arrange() would be less work for the functions to do because it removes numbers then arranges.

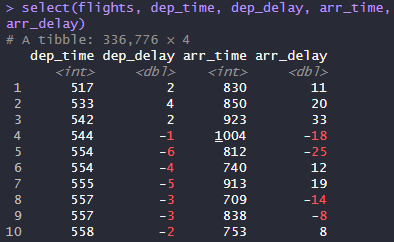
3.3.5 Exercises

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

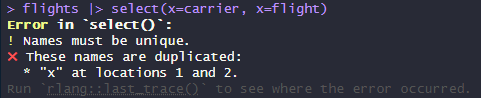


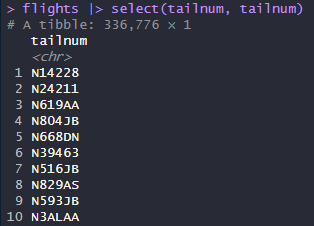
1. Brainstorm as many ways as possible to select dep\_time, dep\_delay, arr\_time, and arr\_delay from flights.





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





1. 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?





The any\_of() function also matches variable names in a character vector, and no errors are thrown for names that don’t exist. It would be helpful to use in conjunction with this vector because it wouldn’t show any errors for missing values.

1. 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?

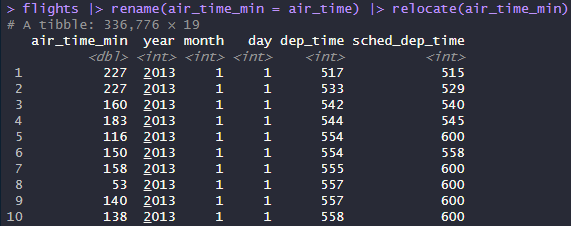


Yes, the result of running the following code does surprise me. The select helpers identify “time” whether it is upper or lower case.

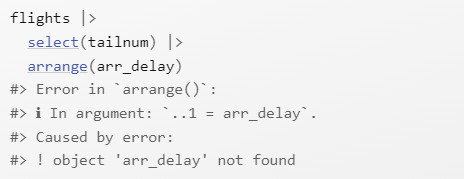




1. Rename air\_time to air\_time\_min to indicate units of measurement and move it to the beginning of the data frame.



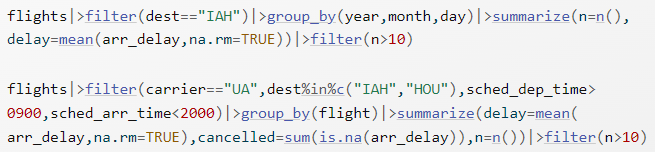
1. Why doesn’t the following work, and what does the error mean?

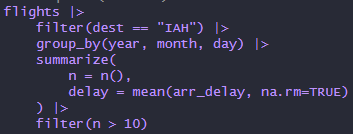


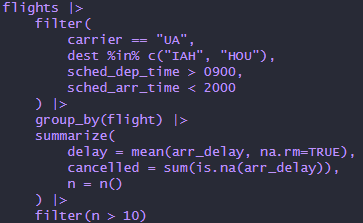
The code doesn’t work because if you want to arrange a column after selecting, that column has to be part of the select(), or the code thinks that the column doesn’t exist.

4.6 Exercises

1. Restyle the following pipelines following the guidelines above.







5.2.1 Exercises

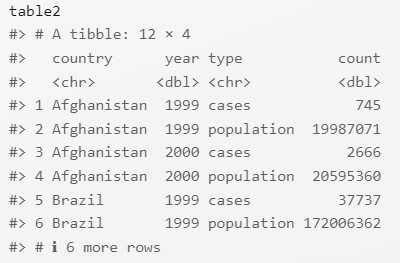
1. For each of the sample tables, describe what each observation and each column represents.

table1



* Each observation represents every country for 1999 and 2000. For example, the first observation is Afghanistan in 1999, and the second observation is Afghanistan in 2000.
* Country represents country, year represents year, cases represents the number of documented cases of tuberculosis, and population represents the population of the country.

table2



* Each observation represents every country for 1999 and 2000 for cases and population. For example, the first observation is Afghanistan in 1999 for cases, and the second observation is Afghanistan in 1999 for population. The next two observations are Afghanistan in 2000 for cases and population.
* Country represents country, year represents year, type represents either cases or population, and count represents the number of cases or population.

table3



* Each observation represents each country for 1999 and 2000 and their rate of tuberculosis. For example, the first observation is Afghanistan in 1999 and the rate of tuberculosis, and the second observation is Afghanistan in 2000 and the rate of tuberculosis.
* Country represents country, year represents year, and rate represents the rate of tuberculosis, which is cases divided by population.

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.

You haven’t yet learned all the functions you’d need to actually perform these operations, but you should still be able to think through the transformations you’d need.

table2:

* Filter as type = “cases” to get TB cases per country per year
* Filter as type = “population” to get the matching population per country per year
* Arrange values so it’s alphabetical by country and oldest year first
* Divide both filters (cases / population) and multiply by 10000
* Create a new column to store values

table3:

* Filter out cases from rate (the numerator of fraction)
* Filter out population from rate (the denominator of fraction)
* Arrange values so it’s alphabetical by country and oldest year first
* Divide both filter (cases / population) and multiply by 10000
* Create a new column to store values