**STOR 320**

**Homework 2**

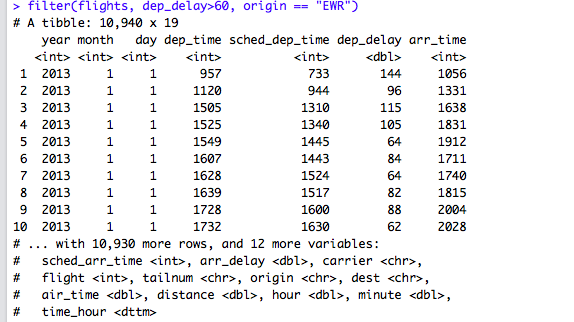
Amanda Liu (PID:730042603)

**1. Class Activity**

1. Determine how many flights out of Newark had departure delays of more than 1 hour. (Use filter, please include either a snapshot, or copy-paste of the code you used here.)

10,940 flights out of Newark had departure delays of more than 1 hour.

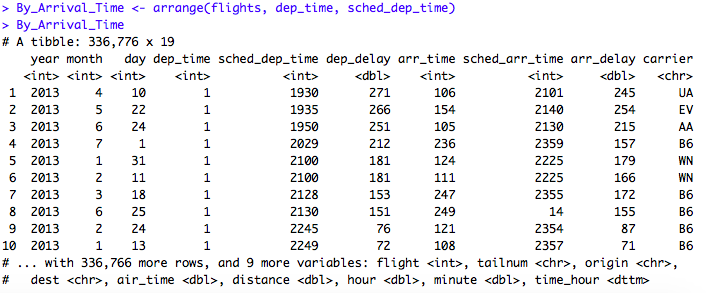
**> filter(flights, dep\_delay>60, origin == "EWR")**

****

1. Re-order (arrange) the rows first by departure time and then scheduled departure time. Save this new table as By\_Arrival\_Time.

**>By\_Arrival\_Time <- arrange(flights, dep\_time, sched\_dep\_time)**

**> By\_Arrival\_Time**



* 1. For the new table, considering the first few rows, why are the scheduled departure times so much larger than the departure times?

Because the scheduled departure time is the day before.

* 1. Order the rows of By\_Arrival\_time by descending order of arrival delay and note the 10th largest arrival delay. How long was this delay in hours?

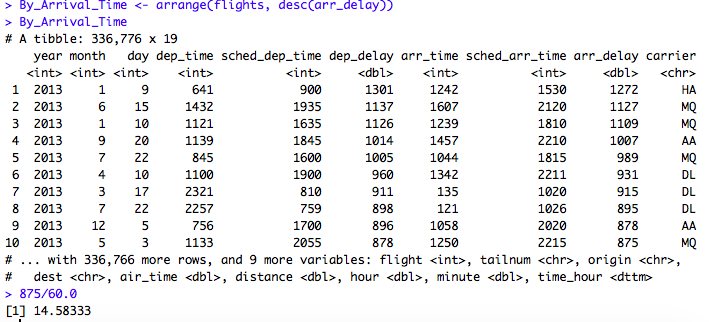
**> By\_Arrival\_Time <- arrange(flights, desc(arr\_delay))**

**> By\_Arrival\_Time**

**> 875/60.0**

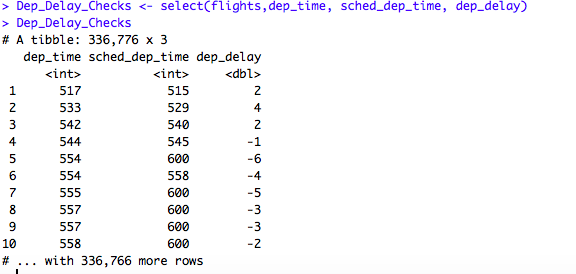
**[1] 14.58333**

This delay was 14.58 hours.



1. Use ‘select’ to make a new table ‘Dep\_Delay\_Checks’ containing only the following variables: dep\_time, sched\_dep\_time, dep\_delay.

**>Dep\_Delay\_Checks<-select(flights,dep\_time,sched\_dep\_time, dep\_delay)**

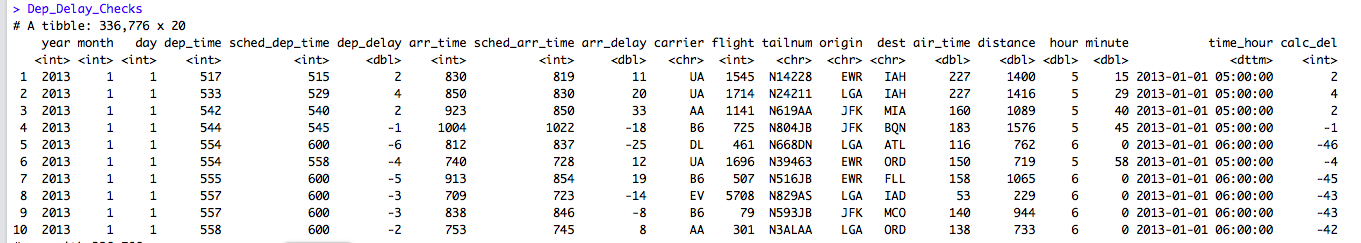
**>Dep\_Delay\_Checks**

* 1. ~~In the new tibble, I notice that the order of variables is different than their order in ‘flights’. Why is this?~~
  2. Use ‘mutate’ to make a new variable ‘calc\_del’ to be calculated from dep\_time and sched\_dep\_time. (s, you will need this column)
     1. Does **mutate** change anything about the original table ‘Dep\_Delay\_Checks’? There are good reasons it does not. To preserve your new column, you can save the table under a new name, or more conveniently, save it as the same name. Do that now.

**flights <- mutate(flights)**

**>Dep\_Delay\_Checks<-mutate(flights, calc\_del = dep\_time - sched\_dep\_time)**

**> Dep\_Delay\_Checks**



* 1. Calc\_del should match another column. A quick glance at the mutated ‘Dep\_Delay\_Checks’ reveals that it does not.
     1. What happened?

Calc\_del does not match dep\_delay. It just calculates the numerical different between dep\_time and sched\_dep\_time, but the time is recorded in different units.

* + 1. We will find a way to deal with this later. For now, we will have to trust the data providers account of dep\_delay and arr\_delay.
  1. Note that you could have made this table with a single command:

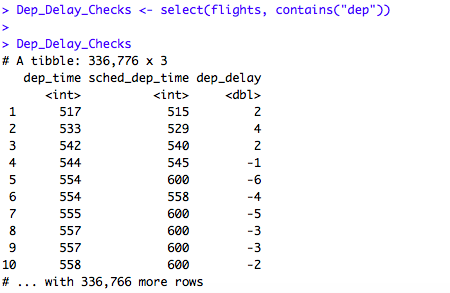
Explain.

1. We would like to associate some carriers with their departure delay times.

**>Dep\_Delay\_Checks <- select(flights, contains(”dep"))**

**>Dep\_Delay\_Checks**

This command create a table only contains variables “dep”, which are dep\_time, sched\_dep\_time and dep\_delay.

****

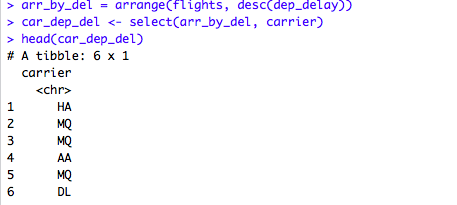
* 1. Using dplyr functions, determine the 3 airlines that had the 3 longest departure delays. (arrange, or perhaps select and arrange)

**> arr\_by\_del = arrange(flights, desc(dep\_delay))**

**> car\_dep\_del <- select(arr\_by\_del, carrier)**

**>head(car\_dep\_del)**

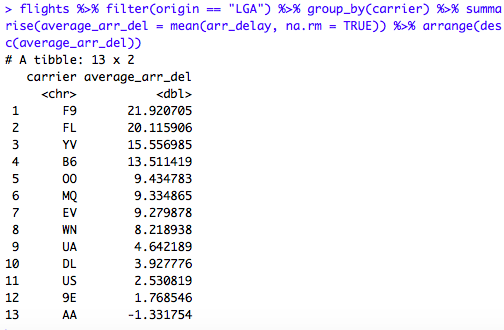
HA,MQ,AA



* 1. Again using dplyr functions, determine the average departure delay and the average arrival delays by carrier (summarise, group\_by … look at the book!)
  2. Flying out of Laguardia, which carriers should you avoid if you hate arrival delays? Explain.

**>flights %>% filter(origin == "LGA") %>% group\_by(carrier) %>% summarise(average\_arr\_del = mean(arr\_delay, na.rm = TRUE)) %>% arrange(desc(average\_arr\_del))**

We should avoid F9 because its arrival delay is longest.



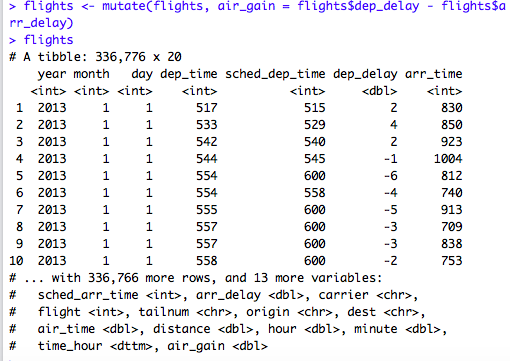
**>flights %>% group\_by(carrier) %>% summarise(ave\_dep\_del = mean(dep\_delay, na.rm = TRUE), ave\_arr\_del = mean(arr\_delay, na.rm = TRUE))**



1. Who is speeding?
   1. Mutate flights to include a column ‘air\_gain’ that shows how much delay time is made up in the air by the airplane. Save this under the same name (flights)

**>flights<-mutate(flights,air\_gain=flights$dep\_delay-flights$arr\_delay)**

**> flights**

****

* 1. Which two airlines have the highest average air\_gain? (Use dplyr functions)

**>average\_air\_gain<-flights%>%group\_by(carrier)%>%summarise(ave\_air\_gain= mean(air\_gain, na.rm=TRUE)) %>% arrange(desc(ave\_air\_gain))**

**> average\_air\_gain\_carrier <- select(average\_air\_gain, carrier)**

**> average\_air\_gain\_carrier**

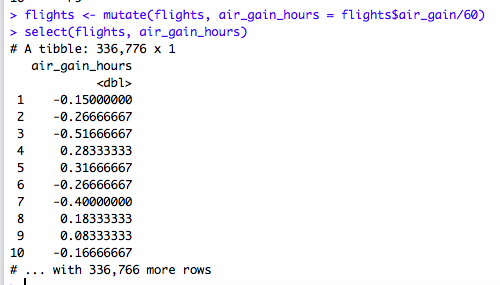
AS, HA

****

* 1. Air\_gain is painful to read as minutes, make a new column air\_gain\_hours.

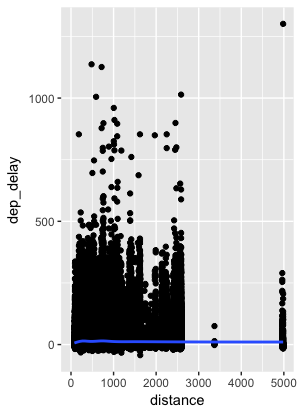
**> flights <- mutate(flights, air\_gain\_hours = flights$air\_gain/60)**

**> select(flights, air\_gain\_hours)**

****

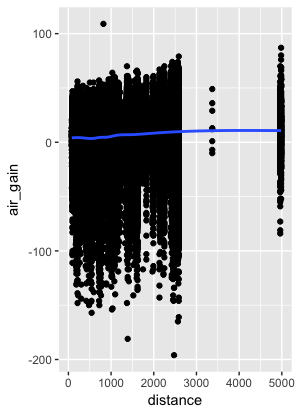
1. Make a plot to get a view of the relationship between
   1. Departure delays and distance

**> ggplot(flights)+geom\_point(mapping = aes(x = distance, y = dep\_delay))+geom\_smooth(mapping = aes(x = distance, y = dep\_delay))**

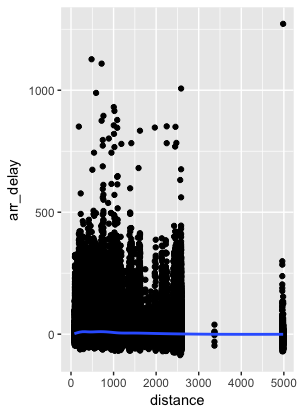
****

* 1. Arrival delays and distance
  2. Air\_gains and distance

**> ggplot(flights)+geom\_point(mapping = aes(x = distance, y = air\_gain))+geom\_smooth(mapping = aes(x = distance, y = air\_gain))**

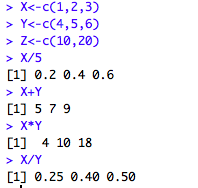
****

**> ggplot(flights)+geom\_point(mapping = aes(x = distance, y = arr\_delay))+geom\_smooth(mapping = aes(x = distance, y = arr\_delay))**

****

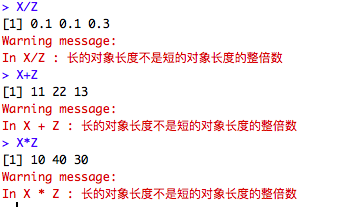
1. Arithmetic on vectors in R (new topic). Enter the following vectors into the console****
   1. Look at X/5, X+Y, X\*Y, X/Y. Which of these do NOT reflect matrix arithmetic? Explain your answer.

X\*Y and X/Y do not reflect matrix arithmetic. Because they are both 3\*1 matrix They cannot be multiplied or divided.

****

* 1. Look at X\*Z, X/Z, X+Z. Explain how R treats arithmetic operations on vectors of different lengths.

If the vectors have different lengths, R will automatically extend the shorter vector by repeating itself.



Predict what X+5 should be and check it in the console.

Each of the component in X will be added in 5. Therefore, the results will be (6,7,8).



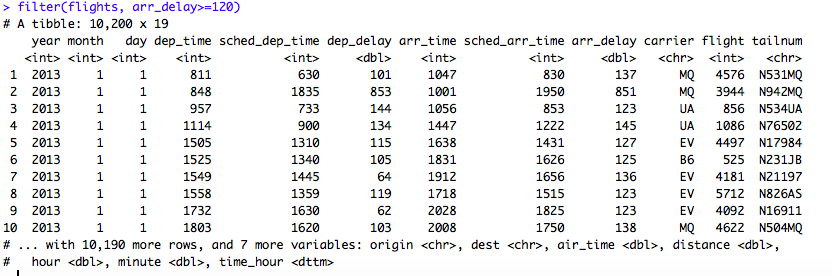
**2. Textbook Homework Problems**

* + 1. #1,2,3 (Part 6 of #1 could be a challenge)

1. Find all flights that

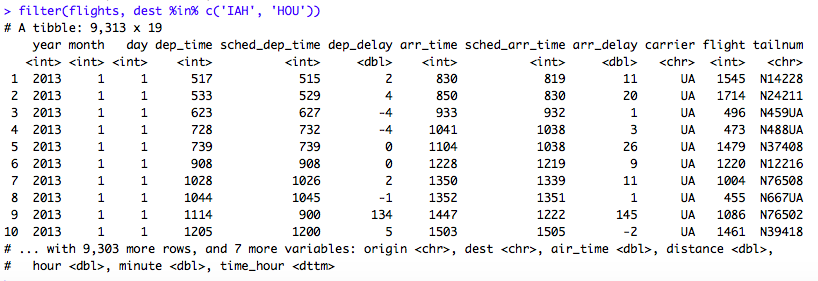
* Had an arrival delay of two or more hours

**> filter(flights, arr\_delay>=120)**

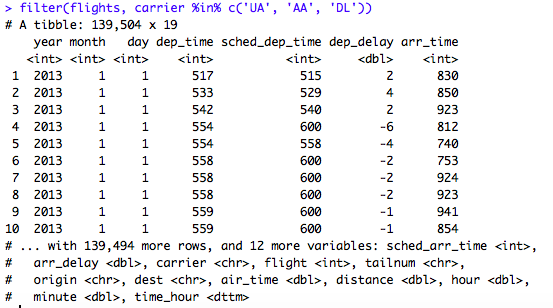


* Flew to Houston (IAH or HOU)
* Were operated by United, American, or Delta

**> filter(flights, dest %in% c('IAH', 'HOU'))**

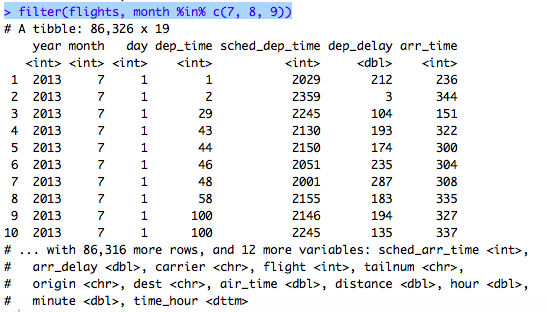


**> filter(flights, carrier %in% c('UA', 'AA', 'DL'))**



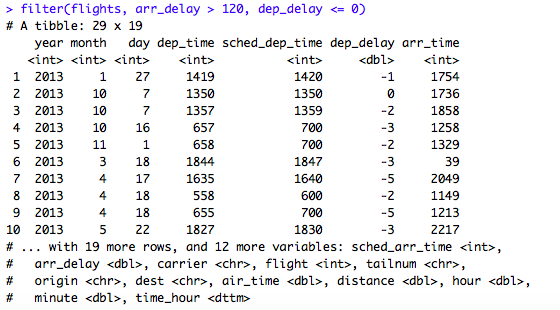
* Departed in summer (July, August, and September)

**> filter(flights, month %in% c(7, 8, 9))**



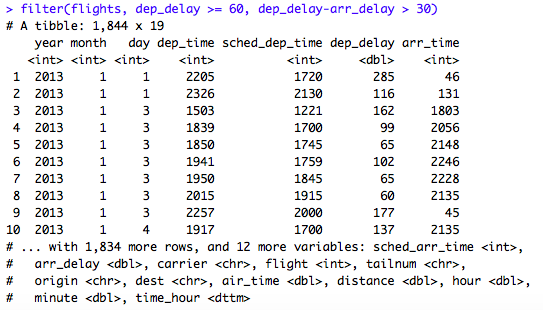
* Arrived more than two hours late, but didn’t leave late

**> filter(flights, arr\_delay > 120, dep\_delay <= 0)**



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

**> filter(flights, dep\_delay >= 60, dep\_delay-arr\_delay > 30)**

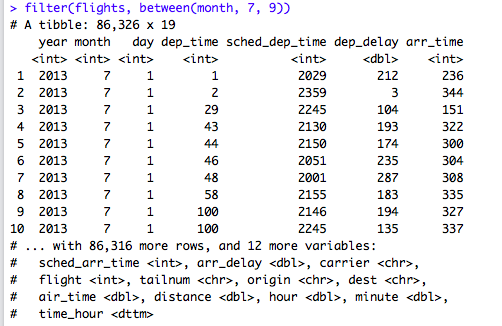


* Departed between midnight and 6am (inclusive)

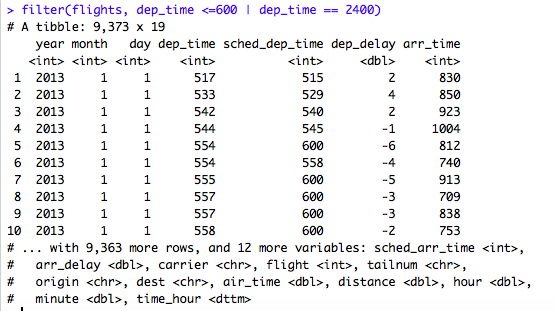
1. Another useful dplyr filtering helper is between(). What does it do? Can you use it to simplify the code needed to answer the previous challenges?

Between is a more convenient way of testing two inequalities at once. It is an easier way to find the results between two values. It tests whether its first is greater or equal to its second, and less or equal to its third. For example, we can simplify the flights in summer by:

**> filter(flights, between(month, 7, 9))**

****

**> filter(flights, dep\_time <=600 | dep\_time == 2400)**



1. How many flights have a missing dep\_time? What other variables are missing? What might these rows represent?
   * 1. #2,3,4

**> sum(is.na(flights$dep\_time))**

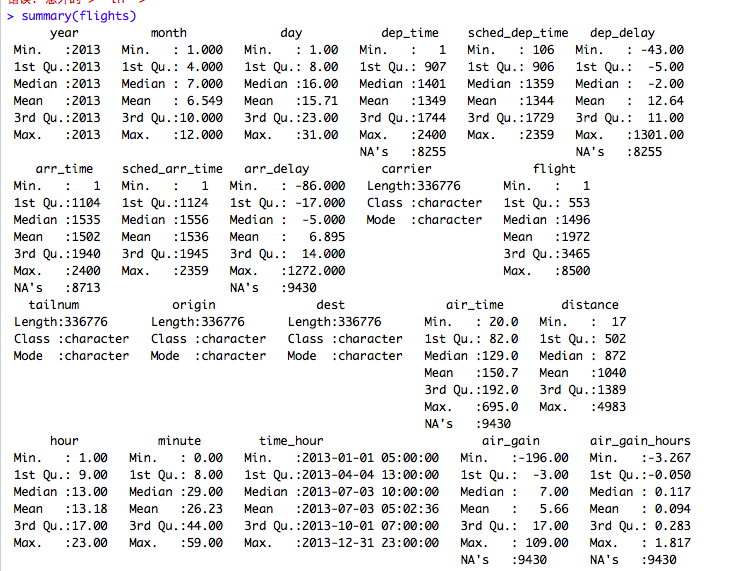
**>summary(flights)**

8255 flights have a missing dep\_time.

arr\_time(8713), dep\_delay(8255), arr\_delay(9430) and air\_time(9430)

It represents that scheduled flights were cancelled or didn’t arrive.

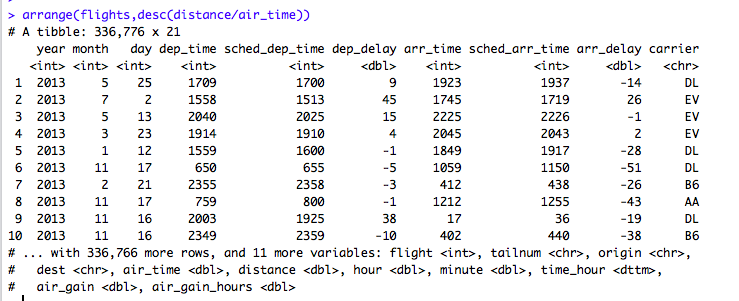


****

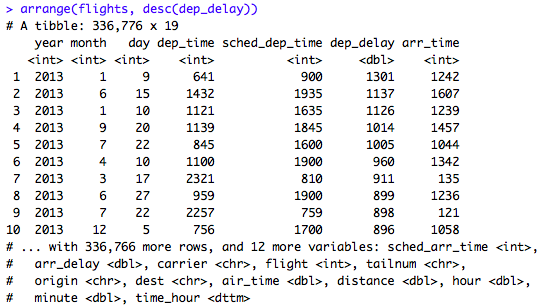
2. Sort flights to find the most delayed flights. Find the flights that left earliest.

1. Sort flights to find the fastest flights.

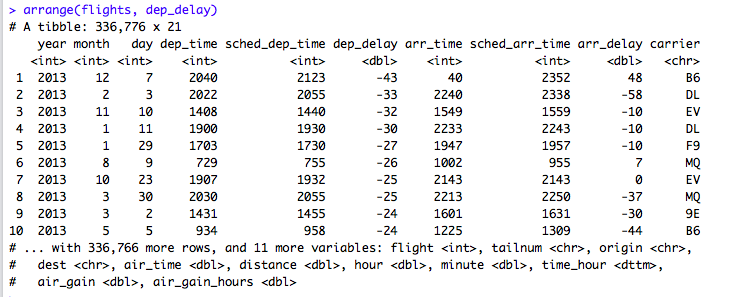
**>arrange(flights,desc(distance/air\_time))**

****

**> arrange(flights, desc(dep\_delay))**

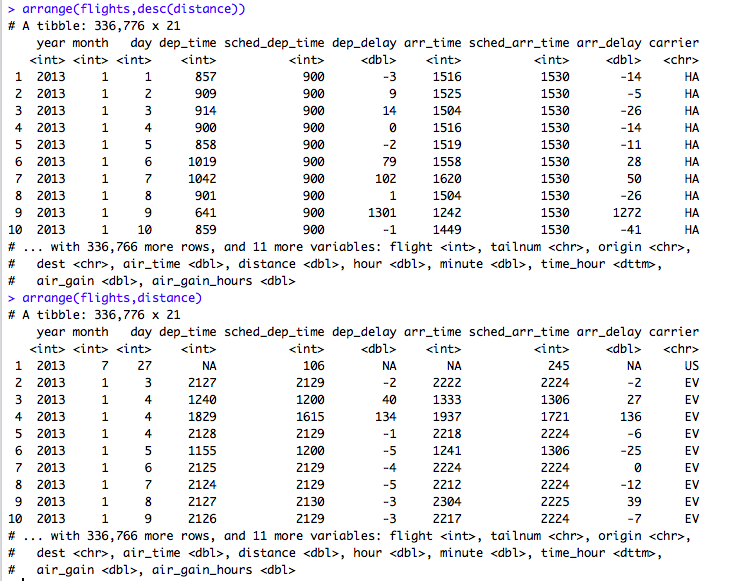


**>arrange(flights, dep\_delay)**

****

1. Which flights travelled the longest? Which travelled the shortest?

Longest: **> arrange(flights,desc(distance))**

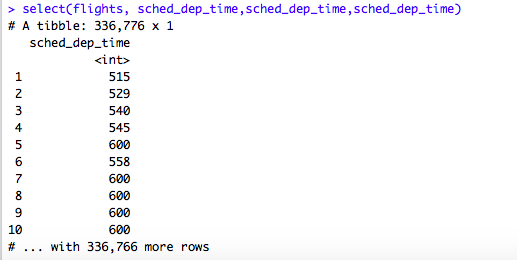
Shortest:**>arrange(flights,distance)**

5.4.1 #2

2. What happens if you include the name of a variable multiple times in a select() call?

**> select(flights, sched\_dep\_time,sched\_dep\_time,sched\_dep\_time)**

Nothing. It still turns out once.



* + 1. #1,2,3,5

1. Currently dep\_time and sched\_dep\_time are convenient to look at, but hard to compute with because they’re not really continuous numbers. Convert them to a more convenient representation of number of minutes since midnight.

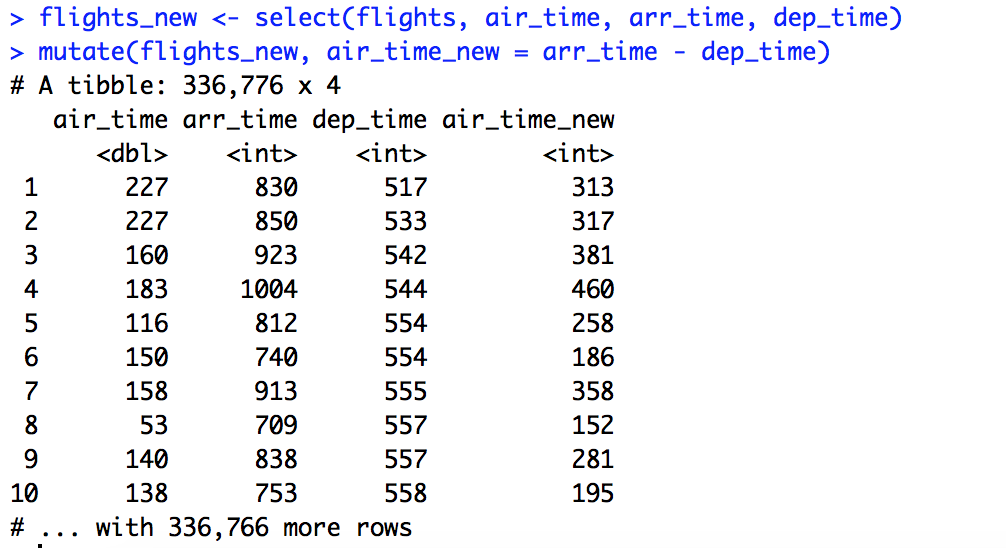
**> transmute(flights, sched\_dep\_time = (sched\_dep\_time%%100)\*60+sched\_dep\_time%%100,dep\_time=(dep\_time%%100)\*60+dep\_time%%100)**

****

1. Compare air\_time with arr\_time - dep\_time. What do you expect to see? What do you see? What do you need to do to fix it?

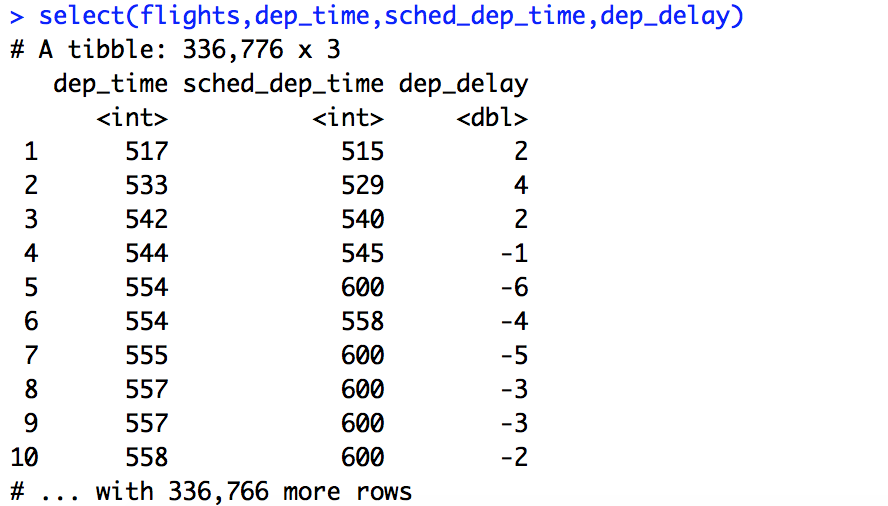
arr\_time is in clock format, but dep\_time is calculated by minutes-after-midnight. So we need to convert arr\_time to minutes-after-midnight.

arr\_time - dep\_time vary significantly from air\_time.

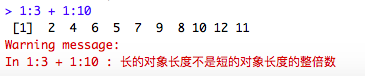


1. Compare dep\_time, sched\_dep\_time, and dep\_delay. How would you expect those three numbers to be related?

We would expect that sched\_dep\_time + dep\_delay == dep\_time.



1. What does 1:3 + 1:10 return? Why?



Because 1:3 is reciclyed. The lengths of these two vectors are not the same, and the shorter one will be automatically extended to be the same length by repeating its elements.