MPA 634  
Data Science for Managers  
Midterm II: Winter 2019

# I. Definitions and Concepts

1. Compare and contrast lists, tibbles, and data frames.  
     
   **list**: In R, vectors are one dimensional holders of information. Vectors have slots or positions in which information and be stored. The information can be referenced by specifying its slot number or name. Although not everyone agrees with Hadley Wickham, he states that there are two kinds of vectors:

* atomic vectors
* lists or recursive vectors

In most programming languages, lists are heterogeneous. This means that they can hold information of a variety of different classes: logical, integer, double, character, etc. The objects in a list can also have a variety of lengths. Lists can even contain other lists. Using lists within lists is a common practice.

**tibble**: A tibble is a collection of atomic vectors that are arranged in a list. Each of the atomic vectors must have the same length in a tibble. This means that tibbles are rectangular lists. This means that a tibble is a special case of a list.

**data frame**: A special type of list used to store data. Each of the atomic vectors in the list must have the same length. This makes the list rectangular. This is the traditional way of storing data in R.

**Compare and contrast lists, tibbles, and data frames**: As already mentioned, data frames and tibbles are both lists. Because all lists do not contain just atomic vectors of the same length, lists in general do not satisfy the requirements for being a tibble. Tibbles are a data frames that has been “tweaked” by Hadley Wickham.

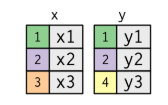
1. Define factors and then explain how they are related to order and level. Use an example to illustrate your answer.

Factor variables usually can have a fixed and known set of values. Factor variables can be represented as characters and integers. An example of a factor would be the grades that are awarded in a class.

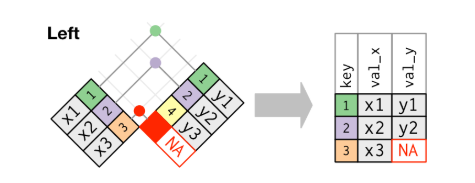
Factors are preferred when a character variable has a reasonably small number of values. Each category in a factor is called a level. Factors have the advantage that they can be ordered.

Factors allow character information to be stored very efficiently on a computer. Each level of a factor can be assigned a code. The code for each observation can then be efficiently substituted for an entire string. Factors are also helpful when we are trying to find data entry errors.

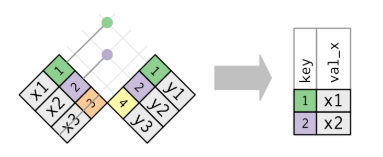
1. Compare and contrast left\_join with semi\_join. Sketch tibbles x and y to illustrate your answer.  
     
   Consider the following two tibbles:



**left\_join**: This is a mutating join since it adds information to tibble x from tibble y wherever the keys match. If there is not match for the primary key value in tibble x with the foreign key value in tibble y, then a missing information NA value is entered.



**semi\_join**: This is a filtering join so it only includes the rows in tibble x that have a match between primary key in x and the foreign key in y. In the above example, this gives the following

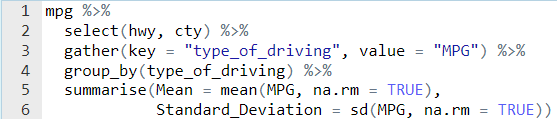


**Comparison and contrast**: left-join is a mutating join so it adds new columns to the x tibble. The semi-join is a filtering join so it takes a subset of the rows of the x tibble that have common key values between the two tibbles.

1. Two classes of time spans are durations and periods. Compare and contrast them.  
     
   **Durations**: Measured as the number of seconds from the origin reference point. For R and Unix, the origin is usually January 1, 1970.  
     
   **Periods**: Measured in human friendly units such as years, months, days, hours, minutes, and seconds.  
     
   **Comparison and contrast**: Periods understand exceptions such as leap years and daylight savings time.
2. Use an example to illustrate the difference between wide and long tibbles.  
     
   

# II. Line by Line Code Interpretation

1. Recall that in the mpg tibble, there are two variables hwy and cty that report highway and city mileage respectively for each vehicle.



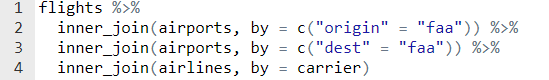
Lines 1 – 2: Select the hwy and cty columns from the mpg tibble

Line 3: Gather converts wide tibbles into a long tibbles. The key is the name of the variable that will be a factor that corresponds to the names of the variables that are being stacked or gathered. In this case the new variable, type\_of\_driving, will either be hwy or cty. The value part of the function specifies the name of the variable where the actual values of the wide variables are stored. In this case, the mileage values are stored in the MPG variable.

Line 4: Declares that the observations should be grouped by the variable type of driving.

Line 5 – 6: Summarise collapses the tibble into two rows, one for the hwy and one for cty. There are two columns in the tibble, Mean and Standard\_Deviation.

1. Recall that flights, airports, and airlines are three different related tibbles that are part of the nycflights13 library.

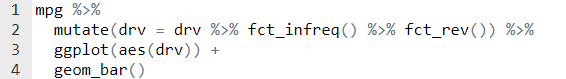


Line 2: Adds information to the flights tibble from the airports tibble where origin in flights matches faa in airports. If there isn’t a match between origin and faa, then the observation in flights is excluded from the new tibble.

Line 3: Adds information to the flights tibble from the airports tibble where dest in flights matches faa in airports. If there isn’t a match between dest and faa, then the observation in flights is excluded from the new tibble.

Line 4: Adds information to the flights tibble from the airlines tibble where carrier in flights matches carrier in lines. If there isn’t a match between the two carrier keys, then the observation in flights is excluded from the new tibble.

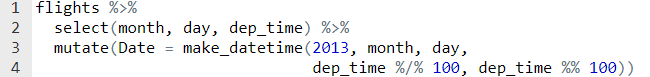
1. Recall that the mpg tibble contains a variable called drv which reports the type of drive train for each vehicle. Vehicles can be front-wheel drive, rear-wheel drive, and four-wheel drive.



Line 2: changes the order of the factor drv. The fct\_infreq() function does a preliminary count of the number of observations in each level of the factor and orders the factor from the largest frequency to the smallest. The fct\_rev reverses the order of the factor from smallest to largest.

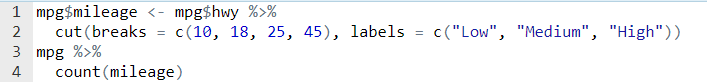
Lines 3 – 4: Create a bar chart. The geom\_bar part of the code recalculates the frequency and orders the bars according to the factor order to create a Pareta Chart

1. The flights tibble contain the variables month, day, and departure time for each flight.

  
  
  
Lines 1 – 2: Select the variables month, day, dep\_time from the tibble flights.

Line 3: Creates a new variable Date using the make\_datetime function The make\_datetime function has as arguments year, month, day, hour, and minutes. Departure time is not in regular time units so we must use integer arithmetic to extract the hour and the minutes from the variable dep\_time. The %/% operator is integer division and gives the result of division rounded down to the nearest integer The %% is the modulus operator and gives the remainder of the integer division.

1. The mpg tibble contains the variable hwy which is miles per gallon for highway driving.



Line 1: Creates a new column in the mpg tibble called mileage

Line 2: Turns the new variable mileage into a factor based on the intervals defined in the cut function. If the hwy variable is greater than 10 and less than or equal to 18, then it is classified as “Low”. If it is greater than 18 but less than or equal to 25, then it is coded as “Medium”. Finally, if the mileage is greater than 25 but less than or equal to 45, then its level is “High”.

Lines 3 – 4: Calculates the frequency of the new factor mileage.