**Exercises**

1. Create three string scalars called "me", "left" and "right". They should contain, respectively, your name, the name of the person sitting to your left and the name of the person sitting to your right. Print these scalars.

* **Tip:** A scalar is a vector with a single element. Check exercise 4 in Lab 1 for some an example of how this can be done.

1. Create three numeric scalars called "me\_sib", "left\_sib" and "right\_sib". They should contain the number of siblings you, the person to your left and the person to your right have, respectively. Print these scalars.

* **Tip:** Check slide 32 in Lab 1 for some an example of how this can be done. If the someone next to you don't have any siblings or you don't have any yourself, create a scalar with NA (no quotes) instead of a number.

1. Create a vector called "age\_young" containing the age of your youngest sibling and of those of the people sitting next to you. If they don't have any siblings, type NA instead of a number.
2. Use the c() function to create a vector called "siblings" with the number of siblings you and the people sitting next to you have (use the scalars you created in ex. 2!). Use the class() function to check what type of data is in that vector.

* **Tip:** Check slide 34 in Lab 1 for some an example of how this can be done.

1. Use the c() function to create a vector called "name" containing your name and those of the people sitting next to you (use the scalars you created in ex. 1!). Use the class() function to check what type of data is in that vector.
2. Use the c() function to create a vector containing the name and number of siblings vector you just created. What type of data is in this vector?
3. Create a data frame names "siblings\_df" with the vectors name, siblings and age\_young. Use the str() command to see what's inside this data frame. You'll notice the name vector is no longer stored as string. We'll see very soon what a factor is and why name is no longer a string variable. But before we get there, simply make it a string using the as.character() function.

* Tip: Check slide 37 in Lab 1 for an example of how to create a data frame.

1. Add a factor variable to your data frame indicating the gender of the youngest siblings.

* Tip: Check slide 51 in Lab 1 for an example of how to create a factor and slide 54 for an example of how to add a new variable to a data frame.

1. Create a boolean variable in your data frame indicating if the youngest siblings in your dataset are above the average siblings age.

* Tip: Check slides 53 to 55 in Lab 1 for examples of how to create Boolean variables.

1. Create a new object with a subsample of your data frame containing only siblings who are below the average age.

* Tip: Slides 38 and 39 contain examples of how to use indexing to select columns and rows in a data frame. Alternatively, you can use the function subset(), as in slide 24.

1. Print summary statistics for both the data frame with your full sample and the one with only siblings who are below the average age.

* Tip: Use the summary() function to do this.

1. To prevent the data.frame() function from turning strings into factors, you can use the argument stringAsFactors. Look at this function's help file to see how that is done.

* Tip: Check slide 57 in Lab 1 for an example of how to use the help function.