1

|  |  |
| --- | --- |
| **Learning objectives:**   1. **Scope of data object/variable values** 2. **Data cleaning and recoding tricks** 3. **Chi square and analysis of variance (ANOVA)** | |
| We will first explore the concept of scope. In programming, scope defines where a variable can be used or accessed. If you create a variable inside a function the variable is defined only inside the function. Once you leave the function, you can't use that variable anymore. Global variables are defined outside of functions, but are accessible both inside and outside of functions. | |
| In R, if you write the code to the right, you are doing the following:   1. You assigning a value of 5 to adata object/variable **a** 2. You are creating a custom function called change. In that function it takes a number as input and adds a value of **a** to it 3. You are calling the function change(), and providing it an argument, 5. The result is printed to the terminal 4. You are trying to use the print() function the value of **x** to the terminal, but generating an error |  |
| The reason you are getting an error is that **x** has no value outside the ‘scope’ of the function. The function’s scope is what happens in the **curly braces of the function**. Outside the braces, **x** has no value. We get a similar kind of error when running the code to the right.  Note that we can print the value of **x** because it is defined outside the function. However, note that print(**x**) yields a value of 6, not 5. When we call change(), **x** is treated as having the value of **x** inside the function.  The thing to remember is that these variables have values that are sensitive to scope; a variable that is defined strictly in a function has that value only in the function. Note that **a** has value inside the function because the value of **a** was set outside and before the function. |  |
| Finally, recall that the <<- assignment operator can be used to change global variables from within functions. In this case, stuff is a global variable, meaning that it has scope everywhere. When we call the function blah(), we are changing the value of stuff within the function. |  |
| Download these data to your computer and save as a csv file:  <https://docs.google.com/spreadsheets/d/1K6smMZfLnK_j_2Fiy1ikVBeK40dmW3IbnP5e_1eoBNM/edit?usp=sharing>  Use the following to import te (with your file path and file name) | |
| Copy, paste and run the code to the right to rename the variables | names(df) <- c("respondent", "happiness","healthg", "friends", "satisfaction","membership","science","age",  "size","weight") |
| Before we get to clearng the data, we’ll look at the apply() function, which can be handy to know for data cleaning when you want to avoid explicit loops (like *for* or *while*). First, let’s make a small matrix. |  |
| Write and run the code to the right. Specified this way, this function goes across the rows and looks at each element. If the element is less than 0, it returns a TRUE. Otherwise it returns FALSE. |  |
| We can use this output to change values in the data. For example, if the condition evaluates as FALSE at a location in the matrix, this would change all negative values to 0 |  |
| Now we can get to cleaning the real data.  We will keep only records that have valid values all the way through. First **turn all values less than 0 into NA.** |  |
| **Q1. Write a two to three sentence explanation about what is happening in this code. You may need to read up on the length() and is.na() functions.** | |
| Write and run the code to the right. **The innermost function is.na() returns TRUE if a value is NA. The any() function returns TRUE if *any* value inside the row is TRUE. The ‘!’ operator reverses any() so that it is FALSE if any value in the row is TRUE.** The apply function (with the 1 argument) goes through each row. The result is that **if any value in any row is NA, then that row is not included in the output file.**  This could be useful if you wanted to delete records with any incomplete data. |  |
| In R, the word ‘factor’ is used to describe [categorical variables](https://en.wikipedia.org/wiki/Categorical_variable). Currently all columns in the data frame are numeric, but some should proba bly be treated as factor variables. This is because the numbers are not really ‘numbers’ or ‘characters’ but category levels. Write and run the code to the right to change the variable from integer to factor. |  |
| **Q2. Write code to convert all of these variables (healthg, friends, satisfaction, membership, science,size) into factor variables** | |
| Now we are going to recode values in two of the variables (happiness and healthg). The term recoding typically means changing the values of variables. Let's first identify the unique values in these variables. |  |
| Now we have to temporarily turn both these variables into character variables. This is required because you can’t add to or change factor level directly to a new factor variable |  |
| Copy and paste this code to change to new values:  df$happiness[df$happiness=="1"] <- "1. Very happy"  df$happiness[df$happiness=="2"] <- "2. Quite happy"  df$happiness[df$happiness=="3"] <- "3. Not very happy"  df$happiness[df$happiness=="4"] <- "4. Not at all happy"  df$healthg[df$healthg=="1"] <- "1. Very good"  df$healthg[df$healthg=="2"] <- "2. Good"  df$healthg[df$healthg=="3"] <- "3. Fair"  df$healthg[df$healthg=="4"] <- "4. Poor" | |
| Now convert back into factor variables. |  |
| Now we’ll look at a cross tabulation of these two variables. Cross tabulations are used to look at the frequencies of two or more categorical variables combined. |  |
| Note that response frequencies in the last row of the table are very small (fewer than 5 in each). This can cause some statistical challenges. It may be best to combine the ‘not very happy’ and ‘not at all happy’ into a new variable ‘3. Not very or not at all happy’.  **Q3. Write code to make this change. The resulting table should look like this:** | |
| Now we want to know if the two categorical variables are independent of one another.m For example, is a person that reports poor health equally likely to be very happy, quite happy and not very happy (given the frequencies of those responses)? Or is it possible that people with poor health are more likely to give systematically different responses to the happiness question than people who report better health? We can use a chi-squared test to explore this question. First, let’s look at the expected frequencies. |  |
|  | |
| Above is a table of frequencies we’d expect to see if there was no association between these two categorical variables. **Unfortunately, one cell has a small value (< 5),** which means that statistics of this test are sometimes **unreliable.**  Nevertheless, we can run a weighted chi square test. Write and run the code to the right. |  |
| **Q4. Provide a short written interpretation of this result (no more than two sentences). The following link will give you an example of how to think about and interpret the Chi-squared test:**  <https://www.jmp.com/en_au/statistics-knowledge-portal/chi-square-test/chi-square-test-of-independence.html> | |
| Sometimes we want to measure a continuous variable across categorical variables. For example, how does age (more or less continuous) vary with self reported health (categorical)? You could do a bunch of paired t-tests, but that is time consuming, and statistically dubious.  An alternative is Analysis of Variance (ANOVA). ANOVA is a very broad set of tools used for understanding variability in continuous across categorical variables. You can read up on ANOVA here:  <https://www.jmp.com/en_us/statistics-knowledge-portal/one-way-anova.html> | |
| Let’s first make a boxda plot of age by self reported health |  |
| The dark black lines are the median age for each category. The grey areas are quantiles, with the top part of the box the 75 percentile and the bottom part of the box the 25% percentile. The lines are the maximum and minimum values. | |
| Now we will use ANOVA to determine whether or not the data are consistent with the null hypothesis (the population means in age do not vary by health category). |  |
| **Q5. Provide a brief interpretation of this result. One of the assumptions of ANOVA is that the dependent variable (age in this case) is normally distributed. Is this a reasonable assumption? Justify your answer in no more than 2 sentences total.** | |
| **Q6. Find some data on your own. Ensure that the data has one categorical variable with at least three levels, and one continuous numeric variable. Your null hypothesis is that the numeric variable does not vary across the groups. Use one-way ANOVA or Kruskal-Wallis to analyse these data. In your answer, provide 1) a link to the data 2) a description of the data, 3) the results of your analysis and 4) an interpretation of your results. Make it look nice and pretty, and ensure it is no more than 1 page in length.** | |
| **Save all the code and written answers (as appropriate) for the questions above into a document. Ensure that the document is nice, clean and readable. Save the document in a pdf file.**  **This assignment is due *before* next class. Please submit on avenue before next week’s class.** | |