**Lab 1: Introduction to data: 1 - Language of data**

|  |  |
| --- | --- |
| 1.1 | We’ll practice on the dataset, **email50**, which contains a subset of incoming emails for the first three months of 2012 for a single email account. We’ll examine the structure of this dataset and determine the number of rows (observations) and columns (variables).  Before make first exercise, please install **openintro** and **tidyverse** packages, using ***install.packages()*** function, and load using the ***library()*** function.   1. Take a peek at the email50 dataset using ***head()*** function. 2. Take a peek at the email50 dataset using the ***glimpse()*** function.   ***How many observations and variables are there?*** |
| 1.2 | Recall that the ***glimpse()*** function tells us the number of observations and variables in a dataset as well as the names and types of each column, along with a neatly printed preview of their first few values.  Let’s have another look at the email50 data, so we can practice identifying variable types.  Use the glimpse() function to view the variables in the email50 dataset.  Remember that variable descriptions are available in the help file for the dataset, which we can access by typing ***?email50*** (that is, type a question mark followed by the name of the dataset) in the code box and running it.  ***Review the output to identify each variable as either numerical or categorical, and further as discrete or continuous (if numerical) or ordinal or not ordinal (if categorical).*** |
| 1.3 | Categorical data are often stored as factors in R. Next, we’ll practice working with a factor variable, number, from the email50 dataset. This variable tells us what type of number (none, small, or big) an email contains.  Modify the code below to:   1. Create a new dataset called **email50\_big** that is a subset of the original **email50** dataset containing only emails with "big"numbers. For this, use the ***filter()*** function from **dplyr** package (**dplyr** package is included in **tydiverse** package). 2. This information is stored in the **number** variable.   ***Report the dimensions of*email50\_big*using the glimpse() function again. How many emails contain big numbers?***  *You also can use dim() function to check the dimension of the dataset.* |
| 1.4 | We’ll create a categorical version of the **num\_char** variable in the email50 dataset. **num\_char** is the number of characters in an email, in thousands. This new variable will have two levels ("**below median**" and "**at or above median**") depending on whether an email has less than the median number of characters or equal to or more than that value.  The median marks the 50th percentile, or midpoint, of a distribution, so half of the emails should fall in one category and the other half in the other. We will learn more about the median and other measures of center in the next tutorial in this series.  The **email50**dataset is available in your workspace. Modify the code below to:  Find the median number of characters in emails, save the result to a variable named med\_num\_char.  ***Using mutate(), create a new column called*num\_char\_cat*, which discretizes the*num\_char*variable into "*below median*" or "*at or above median*".***  ***Assign the resulting data frame to a new data frame named*email50\_updated*.***  ***Then, using count(), determine the number of emails in each level of*num\_char\_cat*. Evaluate whether these counts match the expected numbers.***  ***Use the if\_else() function.*** |
| 1.5 | Another common way of creating a new variable based on an existing one is by combining levels of a categorical variable. For example, the **email50** dataset has a categorical variable called **number** with levels "***none***", "***small***", and "***big***", but suppose we’re only interested in whether an email contains a number. Next, we will create a variable containing this information and also visualize it.  In the email50 dataset, the number variable has three levels, "none", "some", and "big". We want to create a new variable (number\_cat) that combines the levels "***small***" and "***big***" into one level named "***yes***", and keeps the "***none***" level separate as a level named "***no***". This should sound similar to how we used the ***if\_else()*** function previously, where the logical condition we want to check is whether the level of number is equal to (==) "***none***".  Use what you know about the ***mutate***() and ***if\_else()*** functions to:  create a new column in **email50**called **number\_cat** which discretizes the number variable into "***no***" and "***yes***".  Assign the resulting data frame to a new data frame named **email50\_updated**.  Run the code provided to visualize the distribution of the **number\_cat** variable.  Use the help functions: ***?ggplot***, and ***?geom\_bar***. |
| 1.6 | Next, we’ll visualize the relationship between two numerical variables from the **email50** dataset, separated by whether or not the email was spam. This means that we will use an aspect of the plot (like color or shape) to identify the levels in the spam variable so that we can compare plotted values between them.  Recall that in the ***ggplot()*** function, the first argument is the dataset, then we map the aesthetic features of the plot to variables in the dataset, and finally the ***geom\_XXX()*** layer informs how data are represented on the plot.  Next, we will make a scatterplot by adding a ***geom\_point()*** layer to the ***ggplot()*** call.  Create a scatterplot of number of exclamation points (**exclaim\_mess**) on the y-axis vs. number of characters (**num\_char**) on the x-axis.  Color points by whether or not the email is spam.  Note that when you first make the plot, you will see a color gradient in place of where you expected to see a legend like before. This gradient appears because spam is stored as a numerical variable (0 = no, 1 = yes). But, we want to use it as a categorical variable in this plot. To do this we can force R to think of spam as a factor, by temporarily converting it to a factor inside the ggplot. To do this we insert the name of the variable we want to convert to a factor (spam) into the factor() function (e.g. factor(spam)). |
| 1.7 | Inspect the **evals**data frame using techniques you learned in previous exercises. Use an approach that shows you how many observations and variables are included in the dataset. This data is in the **openintro** package and we will use functions from the **tidyverse** for our analysis.  Alternatively we can use ***summary()*** function. What are the differences between ***glimpse()*** and ***summary()*** functions. |
| 1.8 | **Recode a variable**  The ***cls\_students*** variable in **evals** tells you the number of students in the class. Suppose instead of the exact number of students, you’re interested in whether the class is   * "***small***" (18 students or fewer), * "***midsize***" (19 - 59 students), or * "***large***" (60 students or more). * Recode the ***cls\_students*** variable into a new variable, ***cls\_type***, using ***mutate()*** + ***case\_when()*** (use ?case\_when() to see the documentation)***.*** This new variable should have three levels as described above. Save the resulting data frame (with the new variable) as **evals**. * What type of variable is cls\_type? |
| 1.9 | Please describe in one paragraph the dataset you choused for course project.  The link to datasets:  <https://www.knowledgehut.com/blog/data-science/data-science-datasets> |
| 1.10 | What specific predictions or classifications are you aiming to make with this dataset? |
| 1.11 | Identify the independent and dependent variables. How do they relate to your predictions or classifications? |
| 1.12 | Can you showcase any existing association between these variables using an R function? |
| 1.13 | Enumerate the types of variables present in your dataset. |
| 1.14 | Using R, provide a basic summary (***summary()*** function) of your dataset." |
| 1.15 | Do you have some variables that need to be recoded/discretizec? |
| 1.16 | Prepare 2-4 slides with your project, including the title, problem, data, summary data, scope and objectives. |