Psych 1420 Lab Assignment 1

Michelle.VanTieghem

1/29/2019

Contents

Welcome to R! In this first assignment, we'‘ll be learning about how to navigate R (and RStudio.) You can use R to do spreadsheet-style table manipulation (like Excel), as well as statistics (like SPSS). While it takes a little bit more time to get the hang of using R, we hope you”ll find it rewarding, and choose to use it in your future research!

In this assignment, we will learn how to load in a data file, inspect the data, and do some brief calculations on that data.

1. R environment

First we want to tell R where to find the data files we”ll be using.

Ask R "what is the current working directory?" by calling the function getwd()

```
getwd()
```

```
## [1] "/Users/michellevantieghem2/Documents/Columbia/R_Stats/ExpMethodsTA/cu-psych-1420/assignment1"
```

‘You should see a file path as the output of getwd(). This tells you the folder on your computer that R is currently "inside".’

Before working with data, we need to read it into R'‘s environment. First, make sure you know where the dataset is saved. It should be in the your current working directory, and saved in .csv format.’

Ask R to show you all the files in your current working directory by calling the function list.files()

```
list.files()
```

```
##  [1] "Assignment1_realversion_whenready.R"
##  [2] "Assignment1.R"
##  [3] "customTests.R"
##  [4] "dependson.txt"
##  [5] "initLesson.R"
##  [6] "LabAssignment1_PDF.pdf"
##  [7] "LabAssignment1_PDF.Rmd"
##  [8] "lesson.yaml"
##  [9] "non_swirl_versions"
## [10] "notes_creating_swirl_lesson"
```

‘You should see a series of file names in the output. Make sure there is a file that ends in .csv, as that'‘s the data file we'‘ll be loading.’

‘We’‘ll use the function read.csv() to read data from this CSV file into R. We need to tell read.csv() which file we want to read data from. We will do this by putting the name of the file inside the parentheses of read.csv() so that it knows where to look for the data.’

‘We also need to tell R to store the info from the file in an R object so we can work with the data. We’‘ll do this using the left-arrow operator, <- , to take the data on the RIGHT side, and save it into the label name on the LEFT side. Then, whenever we want to access the data, we just need to tell R the label name and the data will be there.’

Read this CSV data file into R by entering the following command: IntroSurvey <- read.csv(“class_survey1_spring2019.csv”)’

```
IntroSurvey <- read.csv("../class_survey1_spring2019.csv")
```

‘We just read in the data saved in class_survey1_spring2019.csv, and used the left arrow <- to assign that data to the label IntroSurvey. R is case sensitive, so the label IntroSurvey is not the same as the label introsurvey. Our convention will be to label data frames with capital letters, and variables in lowercase.’

2. Exploring dataframes

‘Next, we’‘ll learn about functions that help you explore your data. Sometimes, you’‘ll use these to make sure your data read in correctly, and sometimes you’‘ll use these simply to inspect your data so you know what’‘s in it.’

‘The first exploring function you’‘ll use is str(). str() tells you the following things about an object:’

‘the type of object that IntroSurvey is, the number of observations and number of variables (dimensions) of IntroSurvey, a list of each variable and its class (interval, factor, numeric, etc.), and for each variable, a list of all values’

‘Enter the following command and inspect the str(IntroSurvey)’

```
str(IntroSurvey)
```

```
## 'data.frame':    36 obs. of  19 variables:
##  $ id    : int  2 3 4 5 6 7 8 9 10 11 ...
##  $ MS1   : int  2 7 2 4 7 6 2 1 1 2 ...
##  $ MS2   : int  2 2 5 6 6 6 5 5 3 1 ...
##  $ MS3   : int  2 4 2 3 3 6 7 2 3 1 ...
##  $ MS4   : int  3 4 4 3 4 5 2 5 2 3 ...
##  $ MS5   : int  2 7 5 6 5 6 3 5 3 6 ...
##  $ MS6   : int  3 3 3 4 2 3 5 5 4 6 ...
##  $ MS7   : int  3 5 6 5 6 4 3 6 2 3 ...
##  $ MS8   : int  3 3 6 6 7 2 1 5 1 6 ...
##  $ MS9   : int  2 6 6 4 6 3 1 1 5 1 ...
##  $ MS10  : int  1 4 6 5 5 6 2 1 1 6 ...
##  $ MS11  : int  6 4 5 6 7 5 7 6 5 6 ...
##  $ MS12  : int  6 4 6 6 5 5 7 3 2 2 ...
##  $ MS13  : int  5 4 7 5 6 5 5 5 5 3 ...
##  $ Gender: Factor w/ 2 levels "Female","Male": 1 1 2 1 1 2 1 1 1 1 ...
##  $ Age   : int  21 20 21 24 20 21 20 21 23 22 ...
##  $ Major : Factor w/ 21 levels "Anthropology / Psychology",..: 12 12 12 13 9 14 13 10 19 2 ...
##  $ School: Factor w/ 4 levels "Barnard","CC",..: 2 2 3 3 2 3 2 3 4 1 ...
##  $ Year  : Factor w/ 5 levels "First-Year","Junior",..: 4 2 2 2 2 5 2 5 3 4 ...
```

‘Now, here are some questions about what str() tells you.’

Does str() tell you the names of the columns in a dataframe? CorrectAnswer: Yes

Does str() tell you the data types of the columns in a dataframe? CorrectAnswer: Yes

Does str() tell you how many rows are in a dataframe? CorrectAnswer: Yes

'The next exploring function you'‘ll use is head(). head() prints out the first few rows of a dataframe, so you can peek at what the data look like.'

'Enter the following command and inspect the head(IntroSurvey)'

```
head(IntroSurvey)
```

```
##   id MS1 MS2 MS3 MS4 MS5 MS6 MS7 MS8 MS9 MS10 MS11 MS12 MS13 Gender Age
## 1  2   2   2   2   3   2   3   3   3   2    1    6    6    5 Female  21
## 2  3   7   2   4   4   7   3   5   3   6    4    4    4    4 Female  20
## 3  4   2   5   2   4   5   3   6   6   6    6    5    6    7   Male  21
## 4  5   4   6   3   3   6   4   5   6   4    5    6    6    5 Female  24
## 5  6   7   6   3   4   5   2   6   7   6    5    7    5    6 Female  20
## 6  7   6   6   6   5   6   3   4   2   3    6    5    5    5   Male  21
##                              Major School     Year
## 1                       Psychology     CC   Senior
## 2                       Psychology     CC   Junior
## 3                       Psychology     GS   Junior
## 4                      Psychology      GS   Junior
## 5                            Psych     CC   Junior
## 6 Psychology & Financial Economics     GS Sophmore
```

'Now, here are some questions about what head() tells you.'

Does head() tell you the names of the columns in a dataframe?

CorrectAnswer: Yes

Does head() tell you the data types of the columns in a dataframe? CorrectAnswer: No

Does head() tell you how many rows are in a dataframe? CorrectAnswer: No

Does head() tell you every entry in the first row of a dataframe? CorrectAnswer: Yes

'The last exploring function you'‘ll learn about today is summary(). summary() prints out summarizing info about each column of a dataframe.'

'Enter the following command and inspect the summary(IntroSurvey)'

```
summary(IntroSurvey)
```

```
##        id             MS1            MS2             MS3       
##  Min.   : 2.00   Min.   :1.00   Min.   :1.000   Min.   :1.000  
##  1st Qu.:10.75   1st Qu.:2.00   1st Qu.:2.000   1st Qu.:2.000  
##  Median :19.50   Median :3.00   Median :4.000   Median :3.000  
##  Mean   :19.50   Mean   :3.50   Mean   :3.861   Mean   :3.222  
##  3rd Qu.:28.25   3rd Qu.:5.25   3rd Qu.:5.250   3rd Qu.:4.000  
##  Max.   :37.00   Max.   :7.00   Max.   :7.000   Max.   :7.000  
##                                                                 
##       MS4             MS5             MS6             MS7       
##  Min.   :2.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:3.000   1st Qu.:3.000   1st Qu.:3.000   1st Qu.:2.000  
##  Median :4.000   Median :5.000   Median :4.000   Median :3.500  
##  Mean   :3.694   Mean   :4.417   Mean   :3.917   Mean   :3.694  
##  3rd Qu.:5.000   3rd Qu.:6.000   3rd Qu.:5.000   3rd Qu.:5.000  
##  Max.   :6.000   Max.   :7.000   Max.   :7.000   Max.   :7.000  
##                                                                 
##       MS8             MS9             MS10            MS11     
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :2.000
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:5.000
##  Median :5.000   Median :4.000   Median :3.500   Median :6.000
##  Mean   :4.278   Mean   :3.444   Mean   :3.556   Mean   :5.583
##  3rd Qu.:6.000   3rd Qu.:5.000   3rd Qu.:5.250   3rd Qu.:6.000
##  Max.   :7.000   Max.   :6.000   Max.   :7.000   Max.   :7.000
##
##       MS12            MS13          Gender        Age
##  Min.   :2.000   Min.   :3.000   Female:25   Min.   :19.00
##  1st Qu.:3.000   1st Qu.:5.000   Male  :11   1st Qu.:20.00
##  Median :4.000   Median :5.000               Median :21.00
##  Mean   :4.333   Mean   :5.306               Mean   :23.94
##  3rd Qu.:6.000   3rd Qu.:6.000               3rd Qu.:24.25
##  Max.   :7.000   Max.   :7.000               Max.   :57.00
##
##                             Major                                School
##  Psychology                    :12   Barnard                         : 1
##  Psychology                    : 4   CC                              :14
##  Undecided                     : 2   GS                              :17
##  Anthropology / Psychology     : 1   School of Professional Studies: 4
##  biology                       : 1
##  English Literature and Language: 1
##  (Other)                       :15
##         Year
##  First-Year: 1
##  Junior    :15
##  Post-Bac  : 2
##  Senior    : 9
##  Sophmore  : 9
##
##
```

'Now, here are some questions about what summary() tells you.'

Does summary() tell you the names of the columns in a dataframe? CorrectAnswer: Yes

Does summary() tell you the data types of the columns in a dataframe? CorrectAnswer: No

Does summary() tell you how many rows are in a dataframe? CorrectAnswer: No

Does summary() tell you every entry in the first row of a dataframe? CorrectAnswer: No

Does summary() tell you the mean value of numeric columns in a dataframe? CorrectAnswer: Yes

3. Accessing variables in dataframe

'Now that we''ve explored the whole dataframe IntroSurvey, let''s look more closely at some of the columns contained in IntroSurvey.'

'Accessing individual pieces of a larger dataframe, whether it be rows, columns, or single values, is called INDEXING. To index a column in a dataframe, we can''t just type the name of the column. We need to pull the column out of the dataframe it''s in, using a $.'

'For example, to look at the Age column in IntroSurvey, we need to type the following: IntroSurvey$Age . In R, you use the $ as you would the / for webpAges within a website, or file paths on a computer. It allows you to index a column that''s stored inside of a dataframe.'

'Enter the following command and inspect the IntroSurvey$Age'

```r
IntroSurvey$Age
```

```
##  [1] 21 20 21 24 20 21 20 21 23 22 21 19 20 19 21 26 19 28 31 20 22 37 21
## [24] 21 57 23 45 27 19 25 20 19 23 25 21 20
```

‘Index any column in IntroSurvey using the dollar sign $ operator.’

```r
'IntroSurvey$id'
```

```
## [1] "IntroSurvey$id"
```

Hint: ‘Enter IntroSurvey$ with the name of any column in IntroSurvey after the $’

Now, use any of the exploration functions you’ve found so far to identify two numeric variables in this data.

‘Index a numeric column in IntroSurvey using the dollar sign $ operator.’

```r
IntroSurvey$MS1
```

```
##  [1] 2 7 2 4 7 6 2 1 1 2 4 4 2 3 6 6 5 1 2 1 1 1 2 2 7 2 5 6 6 4 3 7 1 1 5
## [36] 5
```

Hint: ‘Enter IntroSurvey$ with the name of any numeric column in IntroSurvey after the $. If you are not sure which columns are numeric, enter str(IntroSurvey) and inspect the output for columns that say “int” (for integer) or “num” (for numeric).’

‘Index another numeric column in IntroSurvey using the dollar sign $ operator.’

```r
IntroSurvey$MS2
```

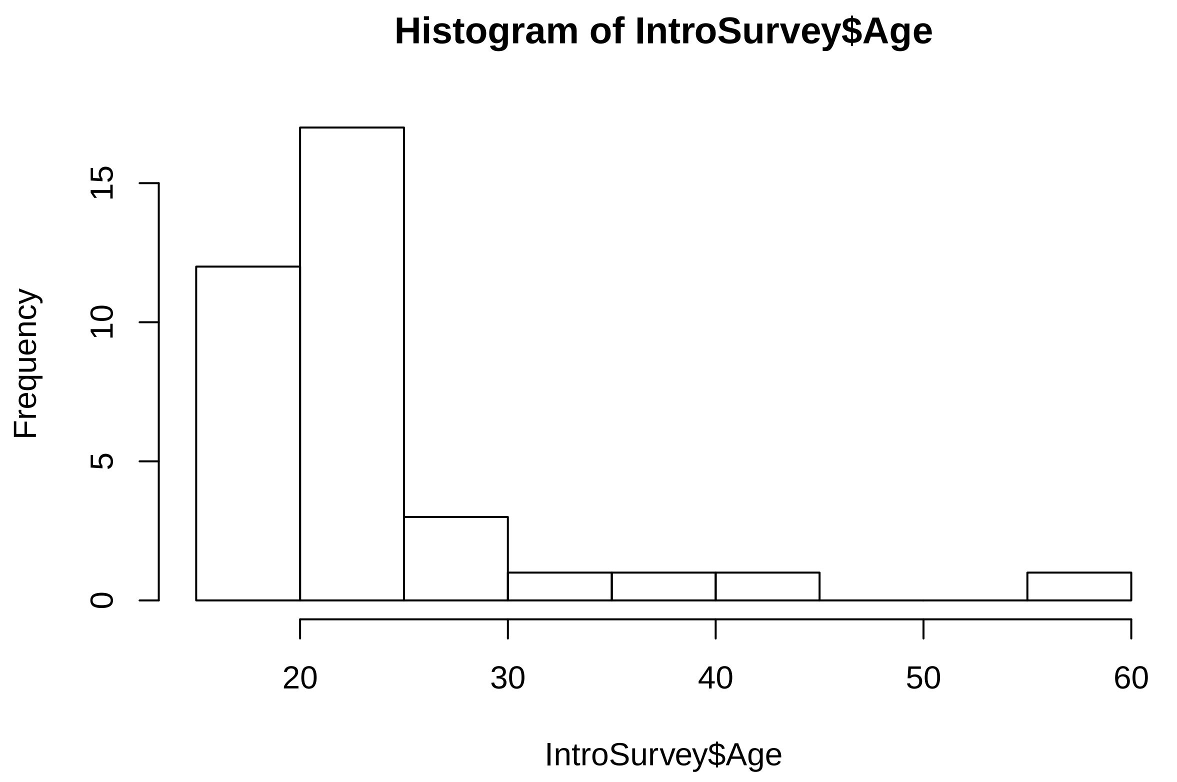
```
##  [1] 2 2 5 6 6 6 5 5 3 1 4 5 2 5 5 6 6 2 3 1 3 3 6 5 1 2 1 4 6 4 2 1 3 5 7
## [36] 6
```

Hint: ‘Enter IntroSurvey$ with the name of any numeric column in IntroSurvey after the $. If you are not sure which columns are numeric, enter str(IntroSurvey) and inspect the output for columns that say “int” (for integer) or “num” (for numeric).’

‘Now, we’‘ll quickly visualize these two numeric columns. Visualizing data in a graph is a great way to quickly inspect it.’

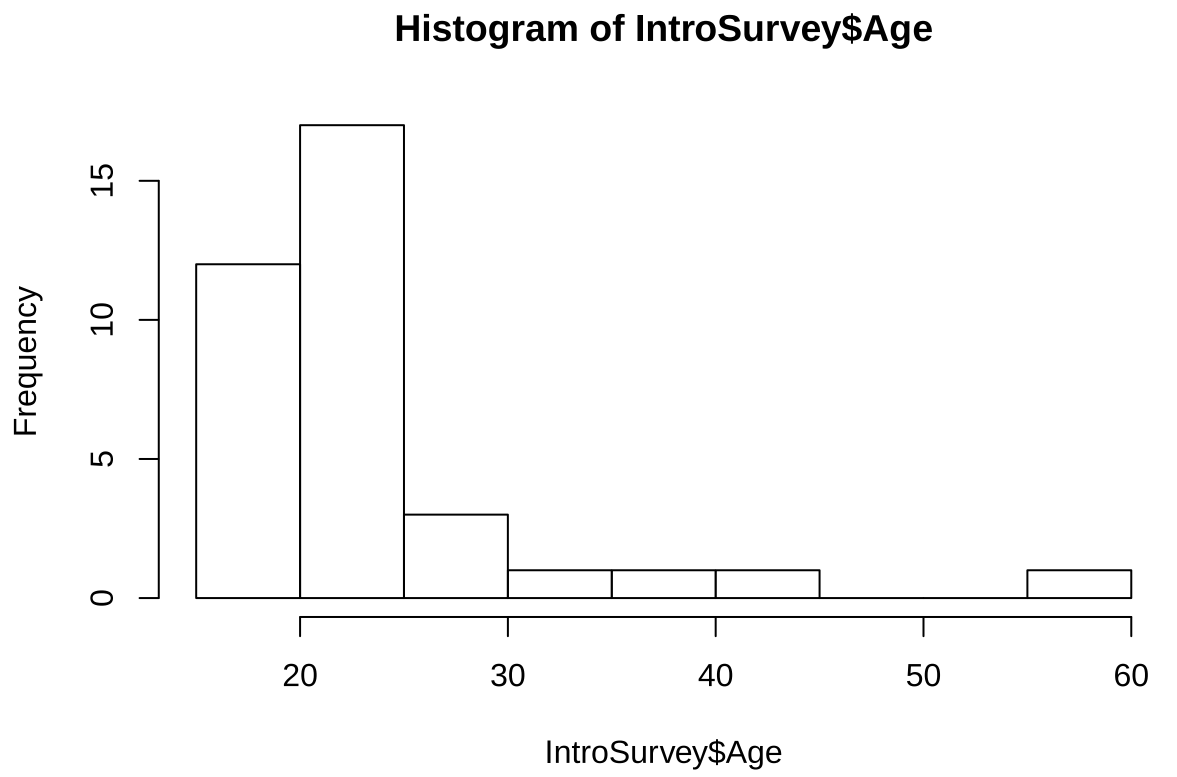
‘Create a quick histogram of your first numeric column in IntroSurvey using hist().’

```r
hist(IntroSurvey$Age)
```



'Create a quick histogram of your second numeric column in IntroSurvey using hist().'

```r
hist(IntroSurvey$Age)
```



Hint: 'Enter hist(IntroSurvey$...), where ... is the name of your numeric column in IntroSurvey, into console and press Enter.'

‘Graphical exploration is one tool you can use to explore the content of specific columns in a dataframe, but it‘’s not the only one. The function summary(), that we used before to explore our whole dataframe, also works on specific columns.’

‘Call summary() on a numeric column in IntroSurvey and inspect the output.’

```r
summary(IntroSurvey$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   19.00   20.00   21.00   23.94   24.25   57.00
```

‘Call summary() on a factor column in IntroSurvey and inspect the output.’

```r
summary(IntroSurvey$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   19.00   20.00   21.00   23.94   24.25   57.00
```

‘Now, let‘’s calculate means for specific columns in our data.’

‘Use the function mean() to find the mean Age of students in the dataframe.’

```r
mean(IntroSurvey$Age)
```

```
## [1] 23.94444
```

‘We now know how to index columns in a dataframe using the $ operator. But what if we want to select just some rows in that column? To index partial columns, we will use hard brackets []. Inside the hard brackets, we will tell R which part of the column we want to index.’

4. Indexing data

‘Usually, when we want to index partial dataframe columns, we only want parts of the column that satisfy some conditions. For example, what if we want to index only the class years for all participants who are older than 21 years? We would use the command IntroSurvey$class[IntroSurvey$Age > 21] . Inside the hard brackets, we have entered a logical statement. This works because in our dataframe, each row contains the data for a single participant. This means that every value of “class” belongs to the same participant as the value of “Age” in the corresponding row of the dataframe. Thus, we can index partial dataframe columns using logical statements about the values of other columns.’

‘For logical statements on numeric columns, we can use the following operators:’

‘== (is equal to), != (is not equal to)’

‘> (greater than), >= (greater than or equal to)’

‘< (less than), <= (less than or equal to)’

‘For logical statements on text columns, we can use the following operators:’

‘== , != . These will check if the string (piece of character data) on the left is equal to the string on the right or not.’

‘For example, IntroSurvey$Major[IntroSurvey$Gender == “Female”] will index all the majors for female-identified students. Notice that for character data, you need to have quotation marks around the data (e.g. “F”) so that R knows that you‘’re referring to character data.’

‘Enter IntroSurvey$Major[IntroSurvey$Gender == “Female”] to see the majors of your female-identifying classmates.’

```r
IntroSurvey$Major[IntroSurvey$Gender == "Female"]
```

```
##  [1] Psychology                      Psychology
##  [3] Psychology                      Psych
##  [5] Psychology                      Psych
##  [7] Psychology Certificate          biology
##  [9] Neuroscience                    psychology
## [11] Psychology                      Psychology
## [13] Psychology                      Anthropology / Psychology
## [15] Graduate Foundations            Psychology and Creative Writing
## [17] Philosophy                      Psychology; Business Management
## [19] Psychology & Statistics         Psychology
## [21] Psychology                      Undecided
## [23] Psychology                      Neuroscience
## [25] Psychology
## 21 Levels: Anthropology / Psychology ... Undecided
```

'Now, we''ll try another one. Use hard brackets and a logical statement to index the school affiliations of all female participants.'

```
IntroSurvey$School[IntroSurvey$Gender == "Female"]
```

```
##  [1] CC                           CC
##  [3] GS                           CC
##  [5] CC                           GS
##  [7] School of Professional Studies Barnard
##  [9] GS                           CC
## [11] CC                           CC
## [13] CC                           GS
## [15] School of Professional Studies CC
## [17] GS                           GS
## [19] GS                           GS
## [21] GS                           GS
## [23] School of Professional Studies CC
## [25] CC
## Levels: Barnard CC GS School of Professional Studies
```

'Now, let''s calculate some descriptive statistics on our data! For the next set of questions, each question will have two parts. The first part will ask you to type in the correct code to calculate a particular descriptive statistic. The second part will ask you to type in the correct value of the descriptive statistic. You can enter the number that is output by the answer to the first part of the question to answer the second part of the question.'

'Enter a command to calculate the mean Age of our participants.'

```
mean(IntroSurvey$Age)
```

```
## [1] 23.94444
```

'Enter a command that tells you which year the participants are in'

```
summary(IntroSurvey$Year)
```

```
## First-Year     Junior   Post-Bac     Senior   Sophmore
##          1         15          2          9          9
```

'Enter a command to calculate the mean Age of the juniors.'

```
mean(IntroSurvey$Age[IntroSurvey$Year == "Junior"])
```

```
## [1] 23.6
```

‘For the last part of this assignment, we’‘ll create a new variable. Sometimes during data analysis, we might want to combine a group of two or more columns in a meaningful way. For example, if we had student’‘s scores for three exams listed in three columns, we might want to have a new column with each student’‘s test average.’

‘In our current data, we are interested in the “Maximizer Scale” (Schwartz et al. 2002) which we will be using in future labs. We will make a new variable that represents the average scores of the 13 individual questions (e.g. “MS1, MS2, MS3”)’

‘To get average Maximizer scores, we can use the function rowMeans() which calculates the mean value for every row of a dataframe. Then, we’‘ll assign the output of rowMeans() to a new column in IntroSurvey’

‘Run the following command to calculate the mean Maximizer Scale score from the 13 questions: IntroSurvey$MS_total < -rowMeans(cbind(IntroSurvey$MS1, IntroSurvey$MS2, IntroSurvey$MS3, IntroSurvey$MS4, IntroSurvey$MS5, IntroSurvey$MS6, IntroSurvey$MS7, IntroSurvey$MS8, IntroSurvey$MS9, IntroSurvey$MS10, IntroSurvey$MS11, IntroSurvey$MS12, IntroSurvey$MS13))’

```r
IntroSurvey$MS_total <- rowMeans(cbind(IntroSurvey$MS1, IntroSurvey$MS2,
                                       IntroSurvey$MS3, IntroSurvey$MS4,
                                       IntroSurvey$MS5, IntroSurvey$MS6,
                                       IntroSurvey$MS7, IntroSurvey$MS8,
                                       IntroSurvey$MS9, IntroSurvey$MS10,
                                       IntroSurvey$MS11, IntroSurvey$MS12,
                                       IntroSurvey$MS13))
```

‘Now names() on the IntroSurvey dataframe, which provides the names of each variable in the dataset. We can check that the new variable is there. MS_total should show up in the names.’

```r
names(IntroSurvey)
```

```
##  [1] "id"       "MS1"      "MS2"      "MS3"      "MS4"      "MS5"
##  [7] "MS6"      "MS7"      "MS8"      "MS9"      "MS10"     "MS11"
## [13] "MS12"     "MS13"     "Gender"   "Age"      "Major"    "School"
## [19] "Year"     "MS_total"
```

```r
mean(IntroSurvey$MS_total)
```

```
## [1] 4.061966
```