Assignment 1 Reference

CU Psych 1491

Contents

Welcome!

Welcome! 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.

getwd() and list.files()

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/mthieu/Repos/cu-psych-1490/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] "1490_R_Project_1_2017.R" "customTests.R"
## [3] "dependson.txt"           "initLesson.R"
## [5] "lesson_markdown.Rmd"     "lesson.yaml"
```

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.

reading in CSV data with read.csv()

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 our CSV data file into R by entering the following command:

```r
IntroSurvey <- read.csv("classdata_2018.csv")
```

We just read in the data saved in `classdata_2018.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.

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.

Data exploration functions

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.)
- for each variable, a list of all values

```r
str(IntroSurvey)
```

```
## 'data.frame':    67 obs. of  37 variables:
##  $ id                  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ CRT1                : int  5 5 10 10 10 5 5 10 10 105 ...
##  $ CRT2                : int  5 5 100 100 100 5 5 100 5 5 ...
##  $ CRT3                : int  47 47 47 47 24 47 47 47 47 47 ...
##  $ CRT_total           : int  3 3 1 1 0 3 3 1 2 2 ...
##  $ maxi1               : int  6 4 4 2 7 4 1 5 7 7 ...
##  $ maxi2               : int  2 2 2 6 5 2 2 1 1 7 ...
##  $ maxi3               : int  2 2 6 6 6 5 2 3 7 6 ...
##  $ maxi4               : int  6 6 6 5 7 6 7 6 7 7 ...
##  $ maxi5               : int  6 4 2 7 3 1 1 4 1 4 ...
##  $ maxi6               : int  5 5 2 5 4 5 3 6 5 3 ...
##  $ regret1             : int  2 4 6 6 2 2 2 5 7 6 ...
##  $ regret2             : int  5 3 2 7 4 2 5 4 6 6 ...
##  $ regret3             : int  5 2 3 6 2 2 5 4 6 7 ...
##  $ regret4             : int  3 2 4 6 3 2 2 3 1 6 ...
##  $ regret5             : int  2 3 2 5 2 2 2 5 6 6 ...
##  $ courses_enrolled    : int  6 5 3 5 5 3 6 1 4 3 ...
##  $ courses_shopped     : int  7 8 2 5 7 4 7 2 4 4 ...
##  $ points_enrolled     : num  15 18 7 16 13 12 16 4 15 12 ...
##  $ time_planning       : Factor w/ 6 levels "1-3 hours","3-5 hours",..: 6 3 3 3 2 1 4 3 2 1 ...
##  $ courses_satisfaction: int  4 4 2 4 3 2 5 5 5 3 ...
##  $ dec_mode            : Factor w/ 4 levels "affect-based decision mode (e.g., \"going with your gut\
##  $ process_regret      : int  1 1 3 1 1 1 1 1 2 1 ...
##  $ outcome_regret      : int  5 1 2 1 1 3 1 1 2 4 ...
##  $ regret_general      : int  3 2 5 2 1 1 4 2 2 4 ...
##  $ maxi_general        : int  5 5 2 5 6 5 6 6 5 4 ...
##  $ psych_courses       : int  4 5 5 7 3 10 4 10 1 5 ...
##  $ age                 : int  23 19 38 22 NA 23 34 28 19 25 ...
##  $ birthyear           : int  1994 1998 1980 1995 NA 1995 1984 1990 1999 1993 ...
##  $ class               : Factor w/ 6 levels "Junior","Post-bac",..: 4 1 4 4 4 4 1 2 5 1 ...
##  $ school              : Factor w/ 6 levels "Barnard","CC",..: 2 2 4 4 4 4 4 6 2 4 ...
##  $ gender              : Factor w/ 3 levels "F","M","Transman": 2 1 1 1 1 1 1 1 2 3 ...
##  $ handed              : Factor w/ 3 levels "A","L","R": 3 3 3 3 1 3 3 3 3 3 ...
##  $ major               : Factor w/ 12 levels "Anthropology",..: 5 5 5 5 5 5 3 5 5 5 ...
##  $ concentration       : Factor w/ 16 levels "","Business Management",..: 10 3 12 8 12 8 8 12 8 8 ..
##  $ reader              : int  0 1 0 0 1 0 1 1 0 0 ...
##  $ programs            : int  1 3 1 1 1 1 1 1 1 1 ...
```

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.

```r
head(IntroSurvey)
```

```
##   id CRT1 CRT2 CRT3 CRT_total maxi1 maxi2 maxi3 maxi4 maxi5 maxi6 regret1
## 1  1    5    5   47         3     6     2     2     6     6     5       2
## 2  2    5    5   47         3     4     2     2     6     4     5       4
## 3  3   10  100   47         1     4     2     6     6     2     2       6
## 4  4   10  100   47         1     2     6     6     5     7     5       6
## 5  5   10  100   24         0     7     5     6     7     3     4       2
## 6  6    5    5   47         3     4     2     5     6     1     5       2
##   regret2 regret3 regret4 regret5 courses_enrolled courses_shopped
## 1       5       5       3       2                6               7
## 2       3       2       2       3                5               8
## 3       2       3       4       2                3               2
## 4       7       6       6       5                5               5
## 5       4       2       3       2                5               7
## 6       2       2       2       2                3               4
##   points_enrolled     time_planning courses_satisfaction
## 1              15 more than 9 hours                    4
## 2              18         5-7 hours                    4
## 3               7         5-7 hours                    2
## 4              16         5-7 hours                    4
## 5              13         3-5 hours                    3
## 6              12         1-3 hours                    2
##                                                                                           dec_mode
## 1                     role-based decision mode (e.g., taking what a Psychology major ought to take)
## 2 calculation-based decision mode (e.g., weighing pros and cons of each course against one another)
## 3                     role-based decision mode (e.g., taking what a Psychology major ought to take)
## 4 calculation-based decision mode (e.g., weighing pros and cons of each course against one another)
## 5 calculation-based decision mode (e.g., weighing pros and cons of each course against one another)
## 6 calculation-based decision mode (e.g., weighing pros and cons of each course against one another)
##   process_regret outcome_regret regret_general maxi_general psych_courses
## 1              1              5              3            5             4
## 2              1              1              2            5             5
## 3              3              2              5            2             5
## 4              1              1              2            5             7
## 5              1              1              1            6             3
## 6              1              3              1            5            10
##   age birthyear  class school gender handed      major
## 1  23      1994 Senior     CC      M      R Psychology
## 2  19      1998 Junior     CC      F      R Psychology
## 3  38      1980 Senior     GS      F      R Psychology
## 4  22      1995 Senior     GS      F      R Psychology
## 5  NA        NA Senior     GS      F      A Psychology
## 6  23      1995 Senior     GS      F      R Psychology
##                           concentration reader programs
## 1                               Pre-Med      0        1
## 2 Business Management, Hispanic Studies      1        3
## 3                            Psychology      0        1
## 4                                  None      0        1
## 5                            Psychology      1        1
## 6                                  None      0        1
```

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

```r
summary(IntroSurvey)
```

```
##        id            CRT1             CRT2             CRT3      
##  Min.   : 1.0   Min.   :  5.00   Min.   :  5.00   Min.   : 3.00  
##  1st Qu.:17.5   1st Qu.:  5.00   1st Qu.:  5.00   1st Qu.:24.00  
##  Median :34.0   Median :  5.00   Median :  5.00   Median :47.00  
##  Mean   :34.0   Mean   : 12.61   Mean   : 40.15   Mean   :39.16  
##  3rd Qu.:50.5   3rd Qu.: 10.00   3rd Qu.:100.00   3rd Qu.:47.00  
##  Max.   :67.0   Max.   :105.00   Max.   :100.00   Max.   :48.00  
##                                                                  
##    CRT_total        maxi1            maxi2            maxi3      
##  Min.   :0.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:1.000   1st Qu.:4.000   1st Qu.:2.000   1st Qu.:2.000  
##  Median :2.000   Median :5.000   Median :5.000   Median :4.000  
##  Mean   :1.821   Mean   :5.045   Mean   :4.313   Mean   :3.925  
##  3rd Qu.:3.000   3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:6.000  
##  Max.   :3.000   Max.   :7.000   Max.   :7.000   Max.   :7.000  
##                                                                 
##      maxi4            maxi5            maxi6           regret1    
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:5.000   1st Qu.:4.000   1st Qu.:3.000   1st Qu.:4.000  
##  Median :6.000   Median :5.000   Median :4.000   Median :5.000  
##  Mean   :5.478   Mean   :4.642   Mean   :4.313   Mean   :5.119  
##  3rd Qu.:6.500   3rd Qu.:6.000   3rd Qu.:5.500   3rd Qu.:6.000  
##  Max.   :7.000   Max.   :7.000   Max.   :7.000   Max.   :7.000  
##                                                                 
##     regret2          regret3          regret4          regret5    
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000  
##  1st Qu.:4.000   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:3.000  
##  Median :5.000   Median :5.000   Median :5.000   Median :5.000  
##  Mean   :4.776   Mean   :4.224   Mean   :4.194   Mean   :4.537  
##  3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:6.000   3rd Qu.:6.000  
##  Max.   :7.000   Max.   :7.000   Max.   :7.000   Max.   :7.000  
##                                                                 
##  courses_enrolled courses_shopped  points_enrolled           time_planning
##  Min.   :1.000    Min.   : 2.000   Min.   : 4.00   1-3 hours        :11
##  1st Qu.:4.000    1st Qu.: 4.000   1st Qu.:12.50   3-5 hours        :22
##  Median :4.000    Median : 6.000   Median :15.00   5-7 hours        :13
##  Mean   :4.224    Mean   : 5.791   Mean   :14.41   7-9 hours        : 7
##  3rd Qu.:5.000    3rd Qu.: 7.000   3rd Qu.:17.00   less than 1 hour : 4
##  Max.   :7.000    Max.   :14.000   Max.   :22.00   more than 9 hours:10
##
##  courses_satisfaction
##  Min.   :2.00
##  1st Qu.:3.00
##  Median :4.00
##  Mean   :3.91
##  3rd Qu.:5.00
##  Max.   :5.00
##
##                                                                                               dec_mo
##  affect-based decision mode (e.g., "going with your gut")                                         : 5
##  calculation-based decision mode (e.g., weighing pros and cons of each course against one another):3
##  role-based decision mode (e.g., taking what a Psychology major ought to take)                    :30
##  rule-based decision mode (e.g., "I'll take whatever seminar Professor X is offering")            :
##
##
##
##  process_regret outcome_regret  regret_general   maxi_general
##  Min.   :1      Min.   :1.000   Min.   :1.000   Min.   :2.000
##  1st Qu.:1      1st Qu.:1.000   1st Qu.:2.000   1st Qu.:4.000
##  Median :2      Median :2.000   Median :3.000   Median :5.000
##  Mean   :2      Mean   :2.239   Mean   :3.134   Mean   :4.642
##  3rd Qu.:3      3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.:5.000
##  Max.   :5      Max.   :5.000   Max.   :6.000   Max.   :6.000
##
##  psych_courses         age          birthyear
##  Min.   : 1.000   Min.   :18.00   Min.   :1959
##  1st Qu.: 2.000   1st Qu.:20.00   1st Qu.:1993
##  Median : 4.000   Median :22.00   Median :1996
##  Mean   : 4.446   Mean   :23.92   Mean   :1994
##  3rd Qu.: 6.000   3rd Qu.:25.00   3rd Qu.:1997
##  Max.   :10.000   Max.   :59.00   Max.   :2000
##  NA's   :2        NA's   :1       NA's   :1
##                                                                    class
##  Junior                                                              :21
##  Post-bac                                                            : 9
##  second semester junior                                              : 1
##  Senior                                                              :29
##  Sophomore                                                           : 6
##  Taking 6 years total to graduate undergrad, currently on the 5th year: 1
##
##                            school        gender   handed
##  Barnard                     : 1   F       :40   A: 5
##  CC                          :40   M       :23   L: 4
##  Continuing Ed.              : 3   Transman: 1   R:58
##  GS                          :19   NA's    : 3
##  School of Professional Studies: 1
##  SPS                         : 3
```

```
##
##                       major                concentration     reader
##  Psychology              :54   None               :34    Min.   :0.0000
##  Neuroscience & Behavior : 2   Psychology         :10    1st Qu.:0.0000
##  Political Science       : 2   Pre-Med            : 6    Median :0.0000
##  Anthropology            : 1                      : 3    Mean   :0.2687
##  Economics               : 1   Business Management: 2    3rd Qu.:1.0000
##  Psychology, Anthropology: 1   Statistics         : 2    Max.   :1.0000
##  (Other)                 : 6   (Other)            :10
##     programs
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :1.000
##  Mean   :1.493
##  3rd Qu.:2.000
##  Max.   :3.000
##
```

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

Indexing: looking at specific columns in a dataframe

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
```

```
##  [1] 23 19 38 22 NA 23 34 28 19 25 20 25 20 22 20 33 20 19 24 19 18 20 21
## [24] 21 20 26 20 21 33 25 21 20 22 19 31 43 23 29 21 22 22 27 21 24 23 28
## [47] 24 25 20 22 20 22 19 34 20 21 32 20 19 22 20 21 59 21 21 21 22
```

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.

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

For example, if we use `str()` again:

```
str(IntroSurvey)
```

```
## 'data.frame':    67 obs. of  37 variables:
##  $ id                  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ CRT1                : int  5 5 10 10 10 5 5 10 10 105 ...
##  $ CRT2                : int  5 5 100 100 100 5 5 100 5 5 ...
##  $ CRT3                : int  47 47 47 47 24 47 47 47 47 47 ...
##  $ CRT_total           : int  3 3 1 1 0 3 3 1 2 2 ...
##  $ maxi1               : int  6 4 4 2 7 4 1 5 7 7 ...
##  $ maxi2               : int  2 2 2 6 5 2 2 1 1 7 ...
##  $ maxi3               : int  2 2 6 6 6 5 2 3 7 6 ...
##  $ maxi4               : int  6 6 6 5 7 6 7 6 7 7 ...
##  $ maxi5               : int  6 4 2 7 3 1 1 4 1 4 ...
##  $ maxi6               : int  5 5 2 5 4 5 3 6 5 3 ...
##  $ regret1             : int  2 4 6 6 2 2 2 5 7 6 ...
##  $ regret2             : int  5 3 2 7 4 2 5 4 6 6 ...
##  $ regret3             : int  5 2 3 6 2 2 5 4 6 7 ...
##  $ regret4             : int  3 2 4 6 3 2 2 3 1 6 ...
##  $ regret5             : int  2 3 2 5 2 2 2 5 6 6 ...
##  $ courses_enrolled    : int  6 5 3 5 5 3 6 1 4 3 ...
##  $ courses_shopped     : int  7 8 2 5 7 4 7 2 4 4 ...
##  $ points_enrolled     : num  15 18 7 16 13 12 16 4 15 12 ...
##  $ time_planning       : Factor w/ 6 levels "1-3 hours","3-5 hours",..: 6 3 3 3 2 1 4 3 2 1 ...
##  $ courses_satisfaction: int  4 4 2 4 3 2 5 5 5 3 ...
##  $ dec_mode            : Factor w/ 4 levels "affect-based decision mode (e.g., \"going with your gut\
##  $ process_regret      : int  1 1 3 1 1 1 1 1 2 1 ...
##  $ outcome_regret      : int  5 1 2 1 1 3 1 1 2 4 ...
##  $ regret_general      : int  3 2 5 2 1 1 4 2 2 4 ...
##  $ maxi_general        : int  5 5 2 5 6 5 6 6 5 4 ...
##  $ psych_courses       : int  4 5 5 7 3 10 4 10 1 5 ...
##  $ age                 : int  23 19 38 22 NA 23 34 28 19 25 ...
##  $ birthyear           : int  1994 1998 1980 1995 NA 1995 1984 1990 1999 1993 ...
##  $ class               : Factor w/ 6 levels "Junior","Post-bac",..: 4 1 4 4 4 4 1 2 5 1 ...
##  $ school              : Factor w/ 6 levels "Barnard","CC",..: 2 2 4 4 4 4 4 6 2 4 ...
##  $ gender              : Factor w/ 3 levels "F","M","Transman": 2 1 1 1 1 1 1 1 2 3 ...
##  $ handed              : Factor w/ 3 levels "A","L","R": 3 3 3 3 1 3 3 3 3 3 ...
##  $ major               : Factor w/ 12 levels "Anthropology",..: 5 5 5 5 5 5 3 5 5 5 ...
##  $ concentration       : Factor w/ 16 levels "","Business Management",..: 10 3 12 8 12 8 8 12 8 8 ..
##  $ reader              : int  0 1 0 0 1 0 1 1 0 0 ...
##  $ programs            : int  1 3 1 1 1 1 1 1 1 1 ...
```

We can now find two numeric columns to index specifically.

```
IntroSurvey$courses_enrolled
```

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

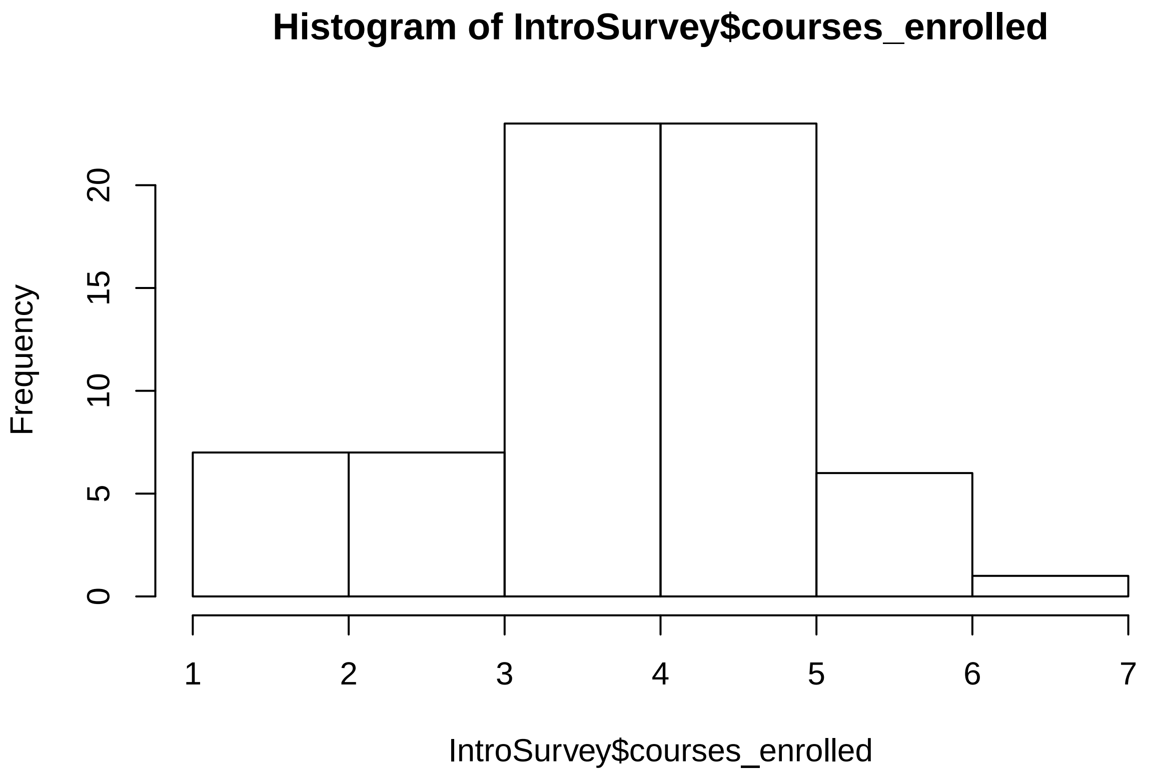
```
IntroSurvey$courses_shopped
```

```
##  [1]  7  8  2  5  7  4  7  2  4  4  9  5  7  7  9  6  6  5  5  4  6 10  5
## [24]  4  7  3  7  5  6  3 10  8  7  5  4  7  4  4  7  8  8  3  4  3  6  2
## [47]  4  4  5  6  7 14  6  4  8  4  7  8  6  4  4  5  4  6  8  5 10
```

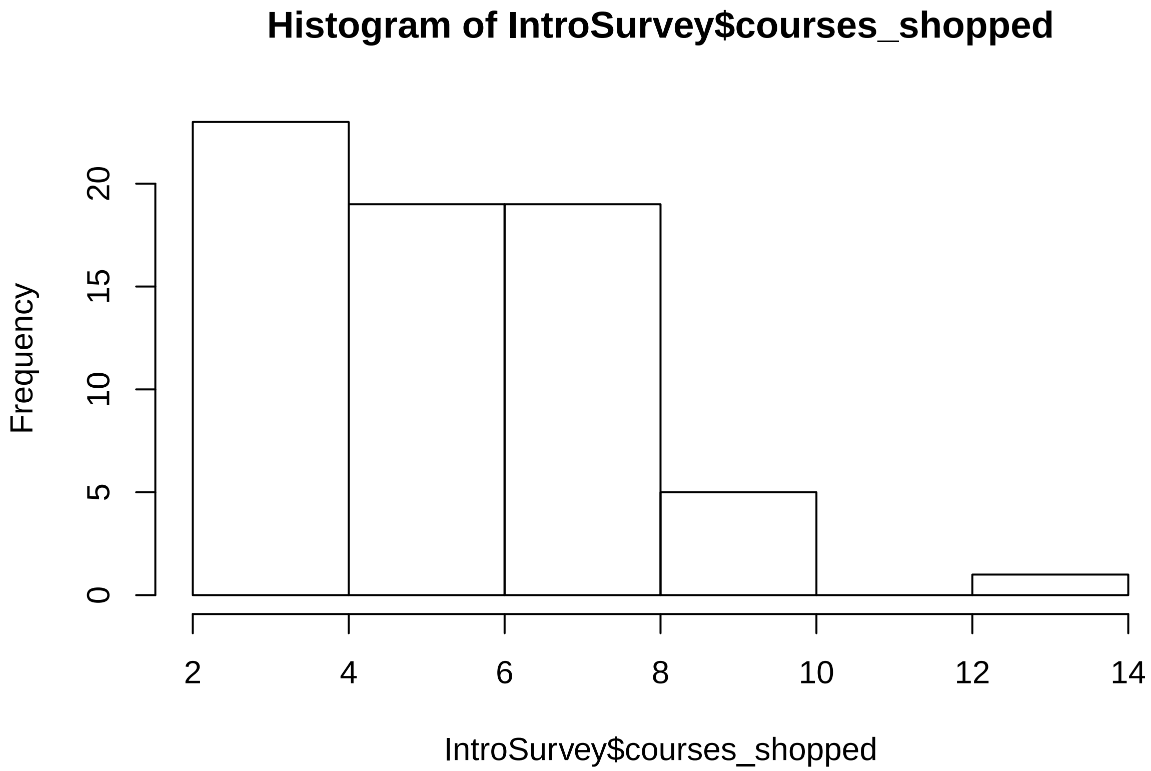
Visualizing data with hist()

Now, we'll quickly visualize these two numeric columns by creating quick histograms with `hist()`. Visualizing data in a graph is a great way to quickly inspect it.

```
hist(IntroSurvey$courses_enrolled)
```



```
hist(IntroSurvey$courses_shopped)
```



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.

If we call it on a numeric column:

```r
summary(IntroSurvey$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   18.00   20.00   22.00   23.92   25.00   59.00       1
```

Versus a factor column:

```r
summary(IntroSurvey$class)
```

```
##                                                                Junior 
##                                                                    21 
##                                                              Post-bac 
##                                                                     9 
##                                                second semester junior 
##                                                                     1 
##                                                                Senior 
##                                                                    29 
##                                                             Sophomore 
##                                                                     6 
## Taking 6 years total to graduate undergrad, currently on the 5th year 
##                                                                     1
```

mean() and missing data

Now, let's find specific descriptive statistics about specific columns in our data.

Use the function mean() to find the mean age of students in the dataframe.

```r
mean(IntroSurvey$age)
```

```
## [1] NA
```

Why does it say `NA`? This means "not available." R is telling you it can't compute a mean. This happens when you have one or more missing values.

Look at the full contents of the age column of IntroSurvey in console to see if there is missing data in the column.

```r
IntroSurvey$age
```

```
##  [1] 23 19 38 22 NA 23 34 28 19 25 20 25 20 22 20 33 20 19 24 19 18 20 21
## [24] 21 20 26 20 21 33 25 21 20 22 19 31 43 23 29 21 22 22 27 21 24 23 28
## [47] 24 25 20 22 20 22 19 34 20 21 32 20 19 22 20 21 59 21 21 21 22
```

You should see that a couple values in `IntroSurvey$age` are not numbers, but `NA`. This means that, for whatever reason in this data, some subjects have missing age data. Rats! `mean()`, and many other functions, will not compute if they see missing values (`NA`). But you can tell them to ignore missing values and compute using the rest of the data. Inside of `mean()`, the argument `na.rm = TRUE` will tell mean() to throw out missing values and compute using the remaining data.

```r
mean(IntroSurvey$age, na.rm = TRUE)
```

```
## [1] 23.92424
```

Partial column indexing with hard brackets []

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.

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?

```
IntroSurvey$class[IntroSurvey$age > 21]
```

```
##  [1] Senior
##  [2] Senior
##  [3] Senior
##  [4] <NA>
##  [5] Senior
##  [6] Junior
##  [7] Post-bac
##  [8] Junior
##  [9] Post-bac
## [10] Senior
## [11] Senior
## [12] Junior
## [13] Post-bac
## [14] Senior
## [15] Post-bac
## [16] second semester junior
## [17] Senior
## [18] Junior
## [19] Post-bac
## [20] Senior
## [21] Senior
## [22] Senior
## [23] Senior
## [24] Senior
## [25] Post-bac
## [26] Post-bac
## [27] Post-bac
## [28] Senior
## [29] Taking 6 years total to graduate undergrad, currently on the 5th year
## [30] Senior
## [31] Post-bac
## [32] Senior
## [33] Senior
## [34] Senior
## [35] Senior
## 6 Levels: Junior Post-bac second semester junior Senior ... Taking 6 years total to graduate undergra
```

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$courses_enrolled[IntroSurvey$gender == "F"]
```

```
##  [1]  5  3  5  5  3  6  1  2  6  5  5 NA  5  4  4  5  5  5  4  2  4  4  5
## [24]  5  4  5  3  2  4  5 NA  4  3  4 NA  5  6  3  5  4  5  2  5
```

The above indexes all the values of enrolled courses 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.

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

```
IntroSurvey$school[IntroSurvey$handed == "L"]
```

```
## [1] CC             Continuing Ed. CC             CC
## 6 Levels: Barnard CC Continuing Ed. GS ... SPS
```

Descriptive stats exercises

Now, let's calculate some descriptive statistics on our data!

What is the mean number of psychology classes taken by our participants?

```
# Be careful of missing data!
mean(IntroSurvey$psych_courses, na.rm = TRUE)
```

```
## [1] 4.446154
```

How many participants are left-handed?

```
summary(IntroSurvey$handed)
```

```
##  A  L  R
##  5  4 58
```

Calling `summary()` on a factor column tells us, in this case, how many students responded "L" for left-handed.

What is the mean age of the juniors?

```
mean(IntroSurvey$age[IntroSurvey$class == "Junior"])
```

```
## [1] 22.33333
```

Creating new dataframe columns with new data

For the last part of this assignment, we'll create new columns in our dataframe for values calculated from existing columns.

For example, you can use this to create columns that contain row-wise means of other columns.

In our dataframe, we will calculate each participant's score on the "Regret Scale" (Schwartz et al., 2002) by averaging the scores on each of 5 different questions: `regret1`, `regret2`, `regret3`, `regret4`, and `regret5`.

We can use the function `rowMeans()` to calculate the mean value for every row of a dataframe. Then, we'll assign the output of `rowMeans()` to a new column in `IntroSurvey` so we can keep everyone's Regret Scale scores with all of their other data.

```
IntroSurvey$regret_total <- rowMeans(cbind(IntroSurvey$regret1,
                                           IntroSurvey$regret2,
                                           IntroSurvey$regret3,
                                           IntroSurvey$regret4,
                                           IntroSurvey$regret5))
```

Now re-run `names()` on the `IntroSurvey` dataframe to check that the new variable is there. `regret_total` should show up in the names.

```
names(IntroSurvey)
```

```
##  [1] "id"                   "CRT1"                 "CRT2"
##  [4] "CRT3"                 "CRT_total"            "maxi1"
##  [7] "maxi2"                "maxi3"                "maxi4"
## [10] "maxi5"                "maxi6"                "regret1"
## [13] "regret2"              "regret3"              "regret4"
## [16] "regret5"              "courses_enrolled"     "courses_shopped"
## [19] "points_enrolled"      "time_planning"        "courses_satisfaction"
## [22] "dec_mode"             "process_regret"       "outcome_regret"
## [25] "regret_general"       "maxi_general"         "psych_courses"
## [28] "age"                  "birthyear"            "class"
## [31] "school"               "gender"               "handed"
## [34] "major"                "concentration"        "reader"
## [37] "programs"             "regret_total"
```

What is the mean regret score for the class?

```
mean(IntroSurvey$regret_total)
```

```
## [1] 4.570149
```

What is the mean regret score for students in Columbia College?

```
mean(IntroSurvey$regret_total[IntroSurvey$CC == "CC"])
```

```
## [1] NaN
```

That concludes the first R assignment. Congratulations!