Intro to Data Science - Lab 2

IST687 Section M003

Professor Anderson

Enter your name here: Hrishikesh Telang

#Select one of the below and add needed information # 1. I did this homework by myself, with help from the book and the professor.

Week 2 – First Breakout: Sorting Data and Ordering a Data Frame

#1. Make a copy of the built-in iris data set like this:

```
myIris <- iris
```

#2. Get an explanation of the contents of the data set with the help function:

```
help("iris")
```

#3. Explore myIris via str and the glimpse functions (note: you need to install and #library 'tidyverse' to use glimpse). Which do you think is better? Why

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

#4. Summarize the variables in your copy of the data set, like this:

```
summary(myIris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
```

```
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

#5. The summary() command provided the mean of each numeric variable. Choose
#the variable with the highest mean and list its contents to the console. Any #variable can be echoed to the console simply by typing its name. Here's an
#example that echoes the variable with the lowest mean to the console

```
myIris$Sepal.Length
```

```
##   [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4 5.1
##  [19] 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.0
##  [37] 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0 6.4 6.9 5.5
##  [55] 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8 6.2 5.6 5.9 6.1
##  [73] 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4 6.0 6.7 6.3 5.6 5.5
##  [91] 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3
## [109] 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7 6.0 6.9 5.6 7.7 6.3 6.7 7.2
## [127] 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8
## [145] 6.7 6.7 6.3 6.5 6.2 5.9
```

#6. Now sort that attribute by calling the sort() function and supplying that variable.
#Remember to choose the variable with the highest mean

```
sort(myIris$Sepal.Length)
```

```
##   [1] 4.3 4.4 4.4 4.4 4.5 4.6 4.6 4.6 4.6 4.7 4.7 4.8 4.8 4.8 4.8 4.8 4.9 4.9
##  [19] 4.9 4.9 4.9 4.9 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.1 5.1 5.1 5.1
##  [37] 5.1 5.1 5.1 5.1 5.1 5.2 5.2 5.2 5.2 5.3 5.4 5.4 5.4 5.4 5.4 5.4 5.5 5.5
##  [55] 5.5 5.5 5.5 5.5 5.5 5.6 5.6 5.6 5.6 5.6 5.6 5.7 5.7 5.7 5.7 5.7 5.7 5.7
##  [73] 5.7 5.8 5.8 5.8 5.8 5.8 5.8 5.8 5.9 5.9 5.9 6.0 6.0 6.0 6.0 6.0 6.0 6.1
##  [91] 6.1 6.1 6.1 6.1 6.1 6.2 6.2 6.2 6.2 6.3 6.3 6.3 6.3 6.3 6.3 6.3 6.3 6.3
## [109] 6.4 6.4 6.4 6.4 6.4 6.4 6.4 6.5 6.5 6.5 6.5 6.5 6.6 6.6 6.7 6.7 6.7 6.7
## [127] 6.7 6.7 6.7 6.7 6.8 6.8 6.8 6.9 6.9 6.9 6.9 7.0 7.1 7.2 7.2 7.2 7.3 7.4
## [145] 7.6 7.7 7.7 7.7 7.7 7.7 7.9
```

#7. Now repeat the previous command, but this time use the order() function, again
#using the variable with the highest mean.

```
order(myIris$Sepal.Length)
```

```
##   [1]  14   9  39  43  42   4   7  23  48   3  30  12  13  25  31  46   2  10
##  [19]  35  38  58 107   5   8  26  27  36  41  44  50  61  94   1  18  20  22
##  [37]  24  40  45  47  99  28  29  33  60  49   6  11  17  21  32  85  34  37
##  [55]  54  81  82  90  91  65  67  70  89  95 122  16  19  56  80  96  97 100
##  [73] 114  15  68  83  93 102 115 143  62  71 150  63  79  84  86 120 139  64
```

```
##  [91]  72  74  92 128 135  69  98 127 149  57  73  88 101 104 124 134 137 147
## [109]  52  75 112 116 129 133 138  55 105 111 117 148  59  76  66  78  87 109
## [127] 125 141 145 146  77 113 144  53 121 140 142  51 103 110 126 130 108 131
## [145] 106 118 119 123 136 132
```

#8. Write a comment in your R code explaining the difference between sort() and #order(). Be prepared to explain this difference to the class

```
#sort() sorts the vector in an ascending order whereas order() returns the
#indices of the vector in a sorted order.
```

#9. Now use the order command to reorder the whole data frame, store the new #dataframe in a variable called 'sortedDF'.

```
sortedDF <- myIris[order(myIris$Sepal.Length),]
View(sortedDF)
```

#10. Now sort the dataframe using arrange(), which is part of the tidyverse package.
#This time, sort based the attribute with the lowest mean. Store the new #dataframe in a variable called 'sortedDF1'

```
sortedDF1 <- myIris %>% arrange(Petal.Width)
sortedDF1
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            4.9         3.1          1.5         0.1     setosa
## 2            4.8         3.0          1.4         0.1     setosa
## 3            4.3         3.0          1.1         0.1     setosa
## 4            5.2         4.1          1.5         0.1     setosa
## 5            4.9         3.6          1.4         0.1     setosa
## 6            5.1         3.5          1.4         0.2     setosa
## 7            4.9         3.0          1.4         0.2     setosa
## 8            4.7         3.2          1.3         0.2     setosa
## 9            4.6         3.1          1.5         0.2     setosa
## 10           5.0         3.6          1.4         0.2     setosa
## 11           5.0         3.4          1.5         0.2     setosa
## 12           4.4         2.9          1.4         0.2     setosa
## 13           5.4         3.7          1.5         0.2     setosa
## 14           4.8         3.4          1.6         0.2     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 16           5.4         3.4          1.7         0.2     setosa
## 17           4.6         3.6          1.0         0.2     setosa
## 18           4.8         3.4          1.9         0.2     setosa
## 19           5.0         3.0          1.6         0.2     setosa
## 20           5.2         3.5          1.5         0.2     setosa
## 21           5.2         3.4          1.4         0.2     setosa
## 22           4.7         3.2          1.6         0.2     setosa
## 23           4.8         3.1          1.6         0.2     setosa
## 24           5.5         4.2          1.4         0.2     setosa
## 25           4.9         3.1          1.5         0.2     setosa
## 26           5.0         3.2          1.2         0.2     setosa
## 27           5.5         3.5          1.3         0.2     setosa
## 28           4.4         3.0          1.3         0.2     setosa
```

```
## 29          5.1         3.4          1.5         0.2     setosa
## 30          4.4         3.2          1.3         0.2     setosa
## 31          5.1         3.8          1.6         0.2     setosa
## 32          4.6         3.2          1.4         0.2     setosa
## 33          5.3         3.7          1.5         0.2     setosa
## 34          5.0         3.3          1.4         0.2     setosa
## 35          4.6         3.4          1.4         0.3     setosa
## 36          5.1         3.5          1.4         0.3     setosa
## 37          5.7         3.8          1.7         0.3     setosa
## 38          5.1         3.8          1.5         0.3     setosa
## 39          5.0         3.5          1.3         0.3     setosa
## 40          4.5         2.3          1.3         0.3     setosa
## 41          4.8         3.0          1.4         0.3     setosa
## 42          5.4         3.9          1.7         0.4     setosa
## 43          5.7         4.4          1.5         0.4     setosa
## 44          5.4         3.9          1.3         0.4     setosa
## 45          5.1         3.7          1.5         0.4     setosa
## 46          5.0         3.4          1.6         0.4     setosa
## 47          5.4         3.4          1.5         0.4     setosa
## 48          5.1         3.8          1.9         0.4     setosa
## 49          5.1         3.3          1.7         0.5     setosa
## 50          5.0         3.5          1.6         0.6     setosa
## 51          4.9         2.4          3.3         1.0 versicolor
## 52          5.0         2.0          3.5         1.0 versicolor
## 53          6.0         2.2          4.0         1.0 versicolor
## 54          5.8         2.7          4.1         1.0 versicolor
## 55          5.7         2.6          3.5         1.0 versicolor
## 56          5.5         2.4          3.7         1.0 versicolor
## 57          5.0         2.3          3.3         1.0 versicolor
## 58          5.6         2.5          3.9         1.1 versicolor
## 59          5.5         2.4          3.8         1.1 versicolor
## 60          5.1         2.5          3.0         1.1 versicolor
## 61          6.1         2.8          4.7         1.2 versicolor
## 62          5.8         2.7          3.9         1.2 versicolor
## 63          5.5         2.6          4.4         1.2 versicolor
## 64          5.8         2.6          4.0         1.2 versicolor
## 65          5.7         3.0          4.2         1.2 versicolor
## 66          5.5         2.3          4.0         1.3 versicolor
## 67          5.7         2.8          4.5         1.3 versicolor
## 68          6.6         2.9          4.6         1.3 versicolor
## 69          5.6         2.9          3.6         1.3 versicolor
## 70          6.1         2.8          4.0         1.3 versicolor
## 71          6.4         2.9          4.3         1.3 versicolor
## 72          6.3         2.3          4.4         1.3 versicolor
## 73          5.6         3.0          4.1         1.3 versicolor
## 74          5.5         2.5          4.0         1.3 versicolor
## 75          5.6         2.7          4.2         1.3 versicolor
## 76          5.7         2.9          4.2         1.3 versicolor
## 77          6.2         2.9          4.3         1.3 versicolor
## 78          5.7         2.8          4.1         1.3 versicolor
## 79          7.0         3.2          4.7         1.4 versicolor
## 80          5.2         2.7          3.9         1.4 versicolor
## 81          6.1         2.9          4.7         1.4 versicolor
## 82          6.7         3.1          4.4         1.4 versicolor
```

```
## 83          6.6         3.0          4.4         1.4 versicolor
## 84          6.8         2.8          4.8         1.4 versicolor
## 85          6.1         3.0          4.6         1.4 versicolor
## 86          6.1         2.6          5.6         1.4  virginica
## 87          6.4         3.2          4.5         1.5 versicolor
## 88          6.9         3.1          4.9         1.5 versicolor
## 89          6.5         2.8          4.6         1.5 versicolor
## 90          5.9         3.0          4.2         1.5 versicolor
## 91          5.6         3.0          4.5         1.5 versicolor
## 92          6.2         2.2          4.5         1.5 versicolor
## 93          6.3         2.5          4.9         1.5 versicolor
## 94          6.0         2.9          4.5         1.5 versicolor
## 95          5.4         3.0          4.5         1.5 versicolor
## 96          6.7         3.1          4.7         1.5 versicolor
## 97          6.0         2.2          5.0         1.5  virginica
## 98          6.3         2.8          5.1         1.5  virginica
## 99          6.3         3.3          4.7         1.6 versicolor
## 100         6.0         2.7          5.1         1.6 versicolor
## 101         6.0         3.4          4.5         1.6 versicolor
## 102         7.2         3.0          5.8         1.6  virginica
## 103         6.7         3.0          5.0         1.7 versicolor
## 104         4.9         2.5          4.5         1.7  virginica
## 105         5.9         3.2          4.8         1.8 versicolor
## 106         6.3         2.9          5.6         1.8  virginica
## 107         7.3         2.9          6.3         1.8  virginica
## 108         6.7         2.5          5.8         1.8  virginica
## 109         6.5         3.0          5.5         1.8  virginica
## 110         6.3         2.7          4.9         1.8  virginica
## 111         7.2         3.2          6.0         1.8  virginica
## 112         6.2         2.8          4.8         1.8  virginica
## 113         6.1         3.0          4.9         1.8  virginica
## 114         6.4         3.1          5.5         1.8  virginica
## 115         6.0         3.0          4.8         1.8  virginica
## 116         5.9         3.0          5.1         1.8  virginica
## 117         5.8         2.7          5.1         1.9  virginica
## 118         6.4         2.7          5.3         1.9  virginica
## 119         7.4         2.8          6.1         1.9  virginica
## 120         5.8         2.7          5.1         1.9  virginica
## 121         6.3         2.5          5.0         1.9  virginica
## 122         6.5         3.2          5.1         2.0  virginica
## 123         5.7         2.5          5.0         2.0  virginica
## 124         5.6         2.8          4.9         2.0  virginica
## 125         7.7         2.8          6.7         2.0  virginica
## 126         7.9         3.8          6.4         2.0  virginica
## 127         6.5         3.0          5.2         2.0  virginica
## 128         7.1         3.0          5.9         2.1  virginica
## 129         7.6         3.0          6.6         2.1  virginica
## 130         6.8         3.0          5.5         2.1  virginica
## 131         6.7         3.3          5.7         2.1  virginica
## 132         6.4         2.8          5.6         2.1  virginica
## 133         6.9         3.1          5.4         2.1  virginica
## 134         6.5         3.0          5.8         2.2  virginica
## 135         7.7         3.8          6.7         2.2  virginica
## 136         6.4         2.8          5.6         2.2  virginica
```

```
## 137          6.4         3.2          5.3         2.3  virginica
## 138          7.7         2.6          6.9         2.3  virginica
## 139          6.9         3.2          5.7         2.3  virginica
## 140          7.7         3.0          6.1         2.3  virginica
## 141          6.9         3.1          5.1         2.3  virginica
## 142          6.8         3.2          5.9         2.3  virginica
## 143          6.7         3.0          5.2         2.3  virginica
## 144          6.2         3.4          5.4         2.3  virginica
## 145          5.8         2.8          5.1         2.4  virginica
## 146          6.3         3.4          5.6         2.4  virginica
## 147          6.7         3.1          5.6         2.4  virginica
## 148          6.3         3.3          6.0         2.5  virginica
## 149          7.2         3.6          6.1         2.5  virginica
## 150          6.7         3.3          5.7         2.5  virginica
```

#11. Finally, use View() to examine your reordered data frames and be prepared to #report on the first few rows

```
View(sortedDF)
View(sortedDF1)
```

#End breakout 1

#Breakout 2

#12. What does the following line of code do?

```
myIris [,1]
```

```
##   [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4 5.1
##  [19] 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.0
##  [37] 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0 6.4 6.9 5.5
##  [55] 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8 6.2 5.6 5.9 6.1
##  [73] 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4 6.0 6.7 6.3 5.6 5.5
##  [91] 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3
## [109] 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7 6.0 6.9 5.6 7.7 6.3 6.7 7.2
## [127] 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8
## [145] 6.7 6.7 6.3 6.5 6.2 5.9
```

```
#This line essentially returns the Column values of 'Sepal.length',
#which corresponds to Index 1. This line works the same as myIris[,"Sepal.Length"].
```

#13. What is the difference (if any) between #myIris [,“Sepal.Length”] #myIris$Sepal.Length

```
#They are the same command.
```

#14. Write the R code that outputs the ‘Sepal.Length’ attribute values, using the #select() command.

```
select(myIris, Sepal.Length)
```

```
##     Sepal.Length
## 1            5.1
```

```
## 2           4.9
## 3           4.7
## 4           4.6
## 5           5.0
## 6           5.4
## 7           4.6
## 8           5.0
## 9           4.4
## 10          4.9
## 11          5.4
## 12          4.8
## 13          4.8
## 14          4.3
## 15          5.8
## 16          5.7
## 17          5.4
## 18          5.1
## 19          5.7
## 20          5.1
## 21          5.4
## 22          5.1
## 23          4.6
## 24          5.1
## 25          4.8
## 26          5.0
## 27          5.0
## 28          5.2
## 29          5.2
## 30          4.7
## 31          4.8
## 32          5.4
## 33          5.2
## 34          5.5
## 35          4.9
## 36          5.0
## 37          5.5
## 38          4.9
## 39          4.4
## 40          5.1
## 41          5.0
## 42          4.5
## 43          4.4
## 44          5.0
## 45          5.1
## 46          4.8
## 47          5.1
## 48          4.6
## 49          5.3
## 50          5.0
## 51          7.0
## 52          6.4
## 53          6.9
## 54          5.5
## 55          6.5
```

```
## 56          5.7
## 57          6.3
## 58          4.9
## 59          6.6
## 60          5.2
## 61          5.0
## 62          5.9
## 63          6.0
## 64          6.1
## 65          5.6
## 66          6.7
## 67          5.6
## 68          5.8
## 69          6.2
## 70          5.6
## 71          5.9
## 72          6.1
## 73          6.3
## 74          6.1
## 75          6.4
## 76          6.6
## 77          6.8
## 78          6.7
## 79          6.0
## 80          5.7
## 81          5.5
## 82          5.5
## 83          5.8
## 84          6.0
## 85          5.4
## 86          6.0
## 87          6.7
## 88          6.3
## 89          5.6
## 90          5.5
## 91          5.5
## 92          6.1
## 93          5.8
## 94          5.0
## 95          5.6
## 96          5.7
## 97          5.7
## 98          6.2
## 99          5.1
## 100         5.7
## 101         6.3
## 102         5.8
## 103         7.1
## 104         6.3
## 105         6.5
## 106         7.6
## 107         4.9
## 108         7.3
## 109         6.7
```

```
## 110          7.2
## 111          6.5
## 112          6.4
## 113          6.8
## 114          5.7
## 115          5.8
## 116          6.4
## 117          6.5
## 118          7.7
## 119          7.7
## 120          6.0
## 121          6.9
## 122          5.6
## 123          7.7
## 124          6.3
## 125          6.7
## 126          7.2
## 127          6.2
## 128          6.1
## 129          6.4
## 130          7.2
## 131          7.4
## 132          7.9
## 133          6.4
## 134          6.3
## 135          6.1
## 136          7.7
## 137          6.3
## 138          6.4
## 139          6.0
## 140          6.9
## 141          6.7
## 142          6.9
## 143          5.8
## 144          6.8
## 145          6.7
## 146          6.7
## 147          6.3
## 148          6.5
## 149          6.2
## 150          5.9
```

#15. Create a new column (called 'Ave.Length') in myIris, which, for each row, is the #average of Sepal.Length and Petal.Length.

```
myIris$Ave.Length <- mean(myIris$Sepal.Length)+mean(myIris$Petal.Length)
myIris$Ave.Length
```

```
##   [1] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
##   [9] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
##  [17] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
##  [25] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
##  [33] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
```

```
##  [41] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
##  [49] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
##  [57] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
##  [65] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
##  [73] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
##  [81] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
##  [89] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
##  [97] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
## [105] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
## [113] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
## [121] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
## [129] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
## [137] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
## [145] 9.601333 9.601333 9.601333 9.601333 9.601333 9.601333
```

#16.What does the following line of code do: which.min(myIris$Petal.Length)

```
#It determines the index location of the column at which the Petal Length
#is the smallest in magnitude.
```

#17. Using the code from the previous step, output the row (iris observation) with the #smallest petal length.

```
myIris[which.min(myIris$Petal.Length),]
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species Ave.Length
## 23          4.6         3.6            1         0.2  setosa   9.601333
```

#18. Create a new dataframe, with just the Petal.Length and Petal.Width attributes

```
newDF <- data.frame(myIris$Petal.Length, myIris$Petal.Width)
View(newDF)
```

#19. Create a new dataframe, using the slice() function, with only the first three rows
#in the myIris dataframe.

```
newDF2 <- myIris %>% slice(1:3)
newDF2
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species Ave.Length
## 1          5.1         3.5          1.4         0.2  setosa   9.601333
## 2          4.9         3.0          1.4         0.2  setosa   9.601333
## 3          4.7         3.2          1.3         0.2  setosa   9.601333
```

#20. Create a new dataframe, which is a subset of myIris, that only includes rows
#where Petal.Length is less than 1.4, store in shortPetalDF

```
shortPetalDF <- myIris[myIris$Petal.Length < 1.4,]
shortPetalDF
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species Ave.Length
## 3           4.7         3.2          1.3         0.2  setosa   9.601333
## 14          4.3         3.0          1.1         0.1  setosa   9.601333
## 15          5.8         4.0          1.2         0.2  setosa   9.601333
## 17          5.4         3.9          1.3         0.4  setosa   9.601333
## 23          4.6         3.6          1.0         0.2  setosa   9.601333
## 36          5.0         3.2          1.2         0.2  setosa   9.601333
## 37          5.5         3.5          1.3         0.2  setosa   9.601333
## 39          4.4         3.0          1.3         0.2  setosa   9.601333
## 41          5.0         3.5          1.3         0.3  setosa   9.601333
## 42          4.5         2.3          1.3         0.3  setosa   9.601333
## 43          4.4         3.2          1.3         0.2  setosa   9.601333
```

#21. How many rows are in the shortPetalDF?

```
nrow(shortPetalDF)
```

```
## [1] 11
```

#22. The homework asks you to create a conditional statement with if and else. A
#conditional statement is part of a larger group of specialized commands that
#control the "flow" of a program – what command gets run and when. You can get #help on if, else, and other control words. Add and run these commands:

```
help("if")
help("Control")
```

#Now add and run your first conditional statement:

```
myNumbers <- c(10,20,40,80,100)
if  (sum(myNumbers) >   40) print("The sum is greater than 40.")
```

```
## [1] "The sum is greater than 40."
```