8451demo

We read in the data from the 8451 data set (This is not the same data set from Project 2! It is only intended to give you an idea about how to use basic functions in R!) The `read.csv` function is used to read in a data frame. The variable `myDF` will be a data frame that stores the data.

```r
myDF <- read.csv("/class/datamine/data/8451/The_Complete_Journey_2_Master/5000_transactions.csv")
```

Please give the data frame a minute or two, to load. It is big!

The data frame has 10625553 rows and 9 columns:

```r
dim(myDF)
```

```
## [1] 10625553        9
```

This is the data that describes the first 6 purchases:

```r
head(myDF)
```

```
##   BASKET_NUM HSHD_NUM PURCHASE_ PRODUCT_NUM SPEND UNITS STORE_R WEEK_NUM
## 1         24     1809 03-JAN-16     5817389 -1.50    -1 SOUTH          1
## 2         24     1809 03-JAN-16     5829886 -1.50    -1 SOUTH          1
## 3         34     1253 03-JAN-16      539501  2.19     1 EAST           1
## 4         60     1595 03-JAN-16     5260099  0.99     1 WEST           1
## 5         60     1595 03-JAN-16     4535660  2.50     2 WEST           1
## 6        168     3393 03-JAN-16     5602916  4.50     1 SOUTH          1
##   YEAR
## 1 2016
## 2 2016
## 3 2016
## 4 2016
## 5 2016
## 6 2016
```

Similarly, these are the amounts spent on the first 6 purchases. We use the dollar sign to pull out a specific column of the data and focus (only) on that column.

```r
head(myDF$SPEND)
```

```
## [1] -1.50 -1.50  2.19  0.99  2.50  4.50
```

These first 6 values in the `SPEND` column add up to a total sum of 7.18 (you can check by hand if you like!)

```r
sum(head(myDF$SPEND))
```

```
## [1] 7.18
```

The average of the first 6 values in the `SPEND` column is 1.196667

```r
mean(head(myDF$SPEND))
```

```
## [1] 1.196667
```

The first 100 values in the `SPEND` column are:

```r
head(myDF$SPEND, n=100)
```

```
##   [1] -1.50 -1.50  2.19  0.99  2.50  4.50  3.49  2.79  1.00  9.98  1.29
##  [12]  1.79  3.99  1.00  2.00 10.80  3.49  1.00  3.99  1.88  0.49  2.49
##  [23]  1.99  2.50  1.67  1.99  5.50  7.89  6.49  1.00  2.78  3.69  1.19
##  [34]  0.69  3.00  5.99  8.19  3.49  4.29  5.66  0.99  5.99  0.99  8.11
##  [45] 12.82  7.99  4.19  1.49  4.96  3.49  4.49  2.79  2.99  5.49  3.99
##  [56] 12.00  3.79  0.89  4.99  2.29  1.69  5.78  6.99  2.00  3.89  6.77
##  [67]  2.69  4.99  3.20 14.40  6.93  2.50  1.00  5.98  1.75  1.19  4.25
##  [78]  3.00  1.11  0.98  8.17 13.10 17.98  4.38  5.79  3.59  4.99 11.56
##  [89]  3.42  2.99 17.99  1.50 -0.38  3.14  2.49  3.99  3.39  1.49  0.53
## [100]  1.25
```

Note that, in the line above, we have an "index" at the far left-hand side of the Console. It shows the position of the first value on each line. The values will change, depending on how wide your screen is.

Here is the 1st value in the `SPEND` column:

```
myDF$SPEND[1]
```

```
## [1] -1.5
```

Here is the 22nd value in the `SPEND` column:

```
myDF$SPEND[22]
```

```
## [1] 2.49
```

Here is the 25th value in the `SPEND` column:

```
myDF$SPEND[25]
```

```
## [1] 1.67
```

Here are the last 20 values in the `SPEND` column. (Notice that we changed **`head`** to **`tail`**, since **`tail`** refers to the end rather than the start.)

```
tail(myDF$SPEND, n=20)
```

```
##  [1] 1.79 4.98 3.99 6.87 0.88 0.69 6.99 1.99 1.99 3.49 3.99 3.99 1.00 0.50
## [15] 3.29 4.99 2.00 5.99 1.79 2.00
```

We can load the help menu for a function in `R` by using a question mark before the function name. It takes some time to get familiar with the style of the `R` help menus, but once you get comfortable reading the help pages, they are very helpful indeed!

```
?head
```

We already took an average of the first 6 entries in the `SPEND` column. Now we can take an average of the entire `SPEND` column.

```
mean(myDF$SPEND)
```

```
## [1] 3.59838
```

Again, here are the first six entries in the `SPEND` column.

```
head(myDF$SPEND)
```

```
## [1] -1.50 -1.50  2.19  0.99  2.50  4.50
```

Suppose that we want to see which entires are bigger than 2 and which ones are smaller than 2. Here are the first six results:

```
head(myDF$SPEND > 2)
```

```
## [1] FALSE FALSE  TRUE FALSE  TRUE  TRUE
```

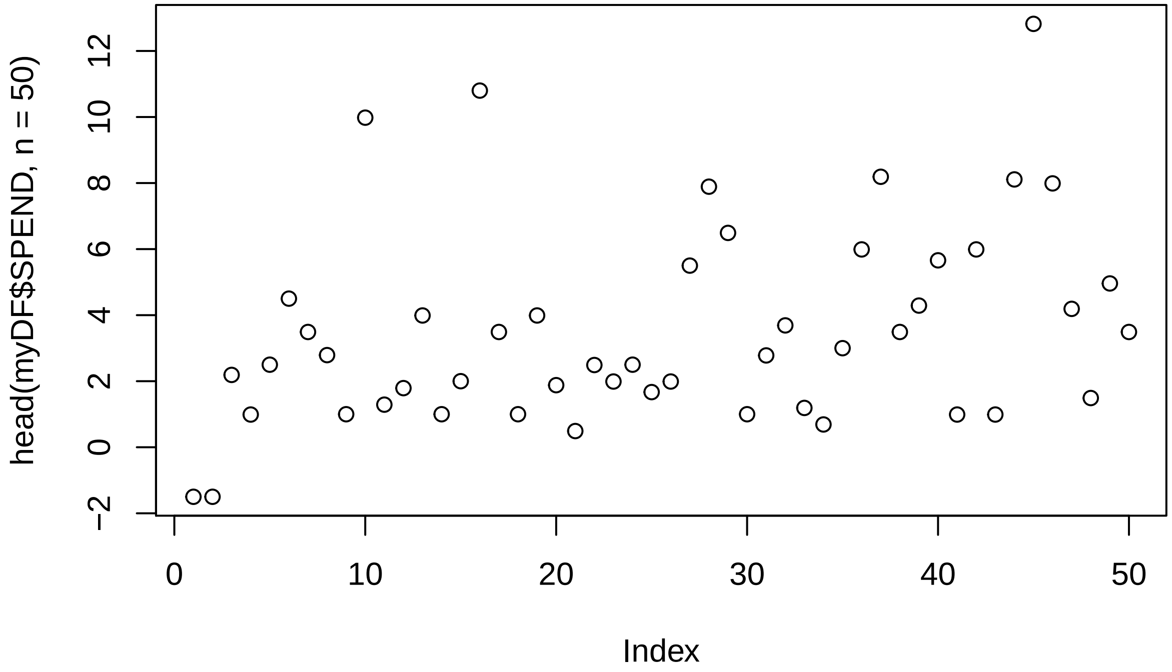
Now we can see what the actual values are. Here are the first 100 such values that are each bigger than 2.

```
head(myDF$SPEND[myDF$SPEND > 2], n=100)
```

```
##   [1]  2.19  2.50  4.50  3.49  2.79  9.98  3.99 10.80  3.49  3.99  2.49
##  [12]  2.50  5.50  7.89  6.49  2.78  3.69  3.00  5.99  8.19  3.49  4.29
##  [23]  5.66  5.99  8.11 12.82  7.99  4.19  4.96  3.49  4.49  2.79  2.99
##  [34]  5.49  3.99 12.00  3.79  4.99  2.29  5.78  6.99  3.89  6.77  2.69
##  [45]  4.99  3.20 14.40  6.93  2.50  5.98  4.25  3.00  8.17 13.10 17.98
##  [56]  4.38  5.79  3.59  4.99 11.56  3.42  2.99 17.99  3.14  2.49  3.99
##  [67]  3.39  8.99  3.34 14.38  5.49  2.47  3.49  5.98  7.99  5.98  5.77
##  [78]  4.00  5.49  3.79  3.34  3.69  2.39 10.00  2.97  5.00  4.79  3.49
##  [89]  5.99  3.99  4.99  3.49  4.54  2.79  2.68  6.78  7.99  3.47  2.69
## [100]  3.49
```

You might want to plot the first 50 values in the SPEND column:

```
plot(head(myDF$SPEND, n=50))
```



If the result says `Error in plot.new() : figure margins too large` then you just need to make your plotting window a little bigger, so that R has room to make the plot, and then run the line again.

There are 10625553 entries in the SPEND column:

```
length(myDF$SPEND)
```

```
## [1] 10625553
```

This makes sense, because the data frame has 10625553 rows and 9 columns.

```
dim(myDF)
```

```
## [1] 10625553        9
```

There are 6322739 entries larger than 2.

```
length(myDF$SPEND[myDF$SPEND > 2])
```

```
## [1] 6322739
```

There are 451155 entries larger than 10.

```
length(myDF$SPEND[myDF$SPEND > 10])
```

```
## [1] 451155
```

There are 4197 entries less than -3.

```
length(myDF$SPEND[myDF$SPEND <= -3])
```

```
## [1] 4197
```

We encourage you to play with the data sets, and to learn how to work with the data, by trying things yourself, and by asking questions. We always welcome your questions, and we love for you to post questions on Piazza. This is a great way for the entire community to learn together!