Project Phoenix learnR Session 2 UNAM

Storing and using data in data.frame

Dr Dave Gillespie - Cardiff University

12 June 2017

Tutorial ideas taken from [www.r-tutor.com/r-introduction/data-frame] and [https://www3.nd.edu/~steve/Rcourse/Lecture2v1.pdf]

Throughout this course, we will be using data.frame to form a structure around our data. A data frame is a list of vectors of the same length. Data frames may contain both numeric and categorical data. Matrices and other data frames can be combined with other data frames, making them a useful tool for manipulating data within R.

Creating a data frame

To create a data frame from scratch, we write:

```
dfexample<-data.frame(gender = c("M","M","F","M","F","F","F","M","M","M"),
                      ht = c(172,186.5,165,180,162.5,179,171,188,175,190),
                      wt = c(91,99,74,80,75,73,87,90,78,73))
```

Try writing this yourself and then writing the *dfexample*. This should give you the following:

```
dfexample
```

```
##    gender    ht wt
## 1       M 172.0 91
## 2       M 186.5 99
## 3       F 165.0 74
## 4       M 180.0 80
## 5       F 162.5 75
## 6       F 179.0 73
## 7       F 171.0 87
## 8       M 188.0 90
## 9       M 175.0 78
## 10      M 190.0 73
```

Note the use of quotation marks to specify categorical variables and how numeric variables are stored using the maximum decimal places specified (i.e. even if *all* entries within a column do not have decimal places specified).

Adding new variables

Adding a new variable is simple. Imagine we also had the age of these individuals. To add these we write:

```
dfexample1<-data.frame(dfexample, age = c(25,18,37,23,27,32,30,32,29,19))
```

which gives us

```
dfexample1
```

```
##    gender    ht wt age
## 1       M 172.0 91  25
## 2       M 186.5 99  18
## 3       F 165.0 74  37
## 4       M 180.0 80  23
## 5       F 162.5 75  27
## 6       F 179.0 73  32
## 7       F 171.0 87  30
## 8       M 188.0 90  32
## 9       M 175.0 78  29
## 10      M 190.0 73  19
```

Note that this vector must be the same length as the other columns in the existing data frame.

Browsing, describing, and summarising your data frame

Browsing your data frame

We have already covered the syntax for viewing the data frame in its entirety. However, in most circumstances this might not be useful. For example, in cases with large data frames containing too many rows or columns to view on one screen. To *preview* a data frame, write the following:

```
head(dfexample1)
```

```
##   gender    ht wt age
## 1      M 172.0 91  25
## 2      M 186.5 99  18
## 3      F 165.0 74  37
## 4      M 180.0 80  23
## 5      F 162.5 75  27
## 6      F 179.0 73  32
```

This provides the first few rows of the data frame. You can also specify the exact number of rows to display. For example, if you want to view the first three rows, write the following:

```
head(dfexample1, n=3)
```

```
##   gender    ht wt age
## 1      M 172.0 91  25
## 2      M 186.5 99  18
## 3      F 165.0 74  37
```

It may also be desirable to browse a particular columns of data. Imagine we wish to look at the "gender" column in our data frame in isolation. This is done by writing the following:

```
dfexample1[1]
```

```
##    gender
## 1       M
## 2       M
## 3       F
## 4       M
## 5       F
## 6       F
## 7       F
## 8       M
## 9       M
## 10      M
```

or writing the specific name of the column with quotation marks:

```
dfexample1["gender"]
```

```
##    gender
## 1       M
## 2       M
## 3       F
## 4       M
## 5       F
## 6       F
## 7       F
## 8       M
## 9       M
## 10      M
```

Columns can also be displayed side-by-side (even if not side-by-side within the whole data frame). We can view the columns containing “gender” and “age” variables, side-by-side, by writing the following:

```
dfexample1[c("gender", "age")]
```

```
##    gender age
## 1       M  25
## 2       M  18
## 3       F  37
## 4       M  23
## 5       F  27
## 6       F  32
## 7       F  30
## 8       M  32
## 9       M  29
## 10      M  19
```

Similarly, we can browse particular rows. In our dataframe, if we want to view row 3 in isoltation we would write:

```
dfexample1[3,]
```

```
##   gender  ht wt age
## 3      F 165 74  37
```

We can view multiple rows (that are not above or below eachother in the original data frame) by writing the following:

```
dfexample1[c(3, 10),]
```

```
##    gender  ht wt age
## 3       F 165 74  37
## 10      M 190 73  19
```

These column and row *slices* can also be combined. For example, say we want to view the "gender" and "age" variables for cases 3 and 10, we would write:

```
dfexample1[c(3,10),c("gender","age")]
```

```
##    gender age
## 3       F  37
## 10      M  19
```

Describing your data frame

You can also describe your data frame using the *str* and *names* commands. Try writing the following:

```
str(dfexample1)
```

```
## 'data.frame':    10 obs. of  4 variables:
##  $ gender: Factor w/ 2 levels "F","M": 2 2 1 2 1 1 1 2 2 2
##  $ ht    : num  172 186 165 180 162 ...
##  $ wt    : num  91 99 74 80 75 73 87 90 78 73
##  $ age   : num  25 18 37 23 27 32 30 32 29 19
```

and

```
names(dfexample1)
```

```
## [1] "gender" "ht"     "wt"     "age"
```

To check the number of rows and columns in your data frame, you can use to *nrow* and *ncol* commands:

```
nrow(dfexample1)
```

```
## [1] 10
```

```
ncol(dfexample1)
```

```
## [1] 4
```

This might be particularly useful when faced with large data frames and needing the check the number of cases and variables.

Summarising your data frame

Another useful function is *summary*, which gives basic summary statistics of each variable in your data frame, tailoring the summary statisics presented depending on the type of variable.

```
summary(dfexample1)
```

```
##  gender       ht              wt             age
##  F:4    Min.   :162.5   Min.   :73.00   Min.   :18.0
##  M:6    1st Qu.:171.2   1st Qu.:74.25   1st Qu.:23.5
##         Median :177.0   Median :79.00   Median :28.0
##         Mean   :176.9   Mean   :82.00   Mean   :27.2
##         3rd Qu.:184.9   3rd Qu.:89.25   3rd Qu.:31.5
##         Max.   :190.0   Max.   :99.00   Max.   :37.0
```

Other summary statistics commands will be covered later on in the course, but this command is always a useful starting point as it gives you a simple check that the variables are stored in the correct way, there are no spurious categories in your categorical variables, and there are no obvious erroneous outliers in your numeric data.

Importing data from another statistics package

Moving on from a simple example to something more closely representing data you might encounter in your day-to-day role, lets import a dataset from SPSS into R, storing it as a data frame. Using the same syntax structure used in the *importing data* session, write:

```
spssexample<-read.spss(file=
                         "data//Simulated data for CH4 030616.sav"
                       , to.data.frame=T, use.value.labels=T)
```

```
## re-encoding from CP1252
```

Note the use of double backslashes when specifying the directory in which your dataset is stored (rather than the single forward slashes we tend to see on web links).

First, lets get a basic description of the structure of our new data frame.

```
str(spssexample)
```

```
## 'data.frame':    100 obs. of  11 variables:
##  $ ID          : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ VAR1        : num  84.9 39 56.8 68.6 46.7 ...
##  $ VAR2        : num  89.9 44 61.8 73.6 51.7 ...
##  $ AVERAGE_V1V2: num  87.4 41.5 59.3 71.1 49.2 ...
##  $ DIFF_V1V2   : num  -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 ...
##  $ ADHERENCE_M1: num  0 1 1 1 0 0 0 1 1 0 ...
##  $ ADHERENCE_M2: num  1 0 1 1 0 0 0 1 1 0 ...
##  $ ADHERENCE_M3: num  13.7 50.8 38.9 87.4 21 ...
##  $ ADHERENCE_M4: num  13.7 50.8 38.9 91.4 21 ...
##  $ AVERAGEM3M4 : num  13.7 50.8 38.9 89.4 21 ...
##  $ DIFFM3M4    : num  0 0 0 -4 0 0 0 0 -6 0 ...
##  - attr(*, "variable.labels")= Named chr
##   ..- attr(*, "names")= chr
##  - attr(*, "codepage")= int 1252
```

Note there are 100 observations and 11 variables. Given this, how might we want to do the following:

1. View our data *(note that 100 observations may be too cumbersome to view on one screen)*
2. Inspect the variables VAR1 and DIFF_V1V2 side-by-side for cases 1 to 5
3. Produce basic summary statistics for the variables in your data frame

Specimen answers

1. View our data *(note that 100 observations may be too cumbersome to view on one screen)*

```
head(spssexample)
```

```
##   ID     VAR1     VAR2 AVERAGE_V1V2 DIFF_V1V2 ADHERENCE_M1 ADHERENCE_M2
## 1  1 84.87767 89.87767     87.37767        -5            0            1
## 2  2 39.04550 44.04550     41.54550        -5            1            0
## 3  3 56.75809 61.75809     59.25809        -5            1            1
## 4  4 68.56952 73.56952     71.06952        -5            1            1
## 5  5 46.74287 51.74287     49.24287        -5            0            0
## 6  6 21.69025 26.69025     24.19025        -5            0            0
##   ADHERENCE_M3 ADHERENCE_M4 AVERAGEM3M4 DIFFM3M4
## 1     13.74859     13.74859    13.74859        0
## 2     50.77670     50.77670    50.77670        0
## 3     38.93857     38.93857    38.93857        0
## 4     87.42027     91.42027    89.42027       -4
## 5     20.99673     20.99673    20.99673        0
## 6     79.44578     79.44578    79.44578        0
```

2. Inspect the variables VAR1 and DIFF_V1V2 side-by-side for cases 1 to 5

```
spssexample[c(1,2,3,4,5),c("VAR1","DIFF_V1V2")]
```

```
##       VAR1 DIFF_V1V2
## 1 84.87767        -5
## 2 39.04550        -5
## 3 56.75809        -5
## 4 68.56952        -5
## 5 46.74287        -5
```

3. Produce basic summary statistics for the variables in your data frame

```
summary(spssexample)
```

```
##        ID              VAR1             VAR2         AVERAGE_V1V2
##  Min.   :  1.00   Min.   :  6.024   Min.   : 11.02   Min.   :  8.524
##  1st Qu.: 25.75   1st Qu.: 35.540   1st Qu.: 40.54   1st Qu.: 38.040
##  Median : 50.50   Median : 47.413   Median : 52.41   Median : 49.913
##  Mean   : 50.50   Mean   : 48.859   Mean   : 53.86   Mean   : 51.359
##  3rd Qu.: 75.25   3rd Qu.: 63.214   3rd Qu.: 68.21   3rd Qu.: 65.714
##  Max.   :100.00   Max.   :110.763   Max.   :115.76   Max.   :113.263
##    DIFF_V1V2   ADHERENCE_M1   ADHERENCE_M2  ADHERENCE_M3
##  Min.   :-5   Min.   :0.00   Min.   :0.0   Min.   : 0.2655
##  1st Qu.:-5   1st Qu.:1.00   1st Qu.:1.0   1st Qu.:21.6819
##  Median :-5   Median :1.00   Median :1.0   Median :42.9640
##  Mean   :-5   Mean   :0.86   Mean   :0.8   Mean   :46.7912
##  3rd Qu.:-5   3rd Qu.:1.00   3rd Qu.:1.0   3rd Qu.:70.5496
##  Max.   :-5   Max.   :1.00   Max.   :1.0   Max.   :99.6612
##   ADHERENCE_M4       AVERAGEM3M4         DIFFM3M4
##  Min.   :  0.2655   Min.   : 0.2655   Min.   :-10.000
##  1st Qu.: 23.5707   1st Qu.:22.3647   1st Qu.: -6.000
##  Median : 48.9706   Median :46.2063   Median : -1.000
##  Mean   : 49.7828   Mean   :48.2870   Mean   : -2.992
##  3rd Qu.: 76.8205   3rd Qu.:73.8543   3rd Qu.:  0.000
##  Max.   :100.0000   Max.   :99.6612   Max.   :  0.000
```