Nunez_Assig1

Jean Jacques

March 14, 2018

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

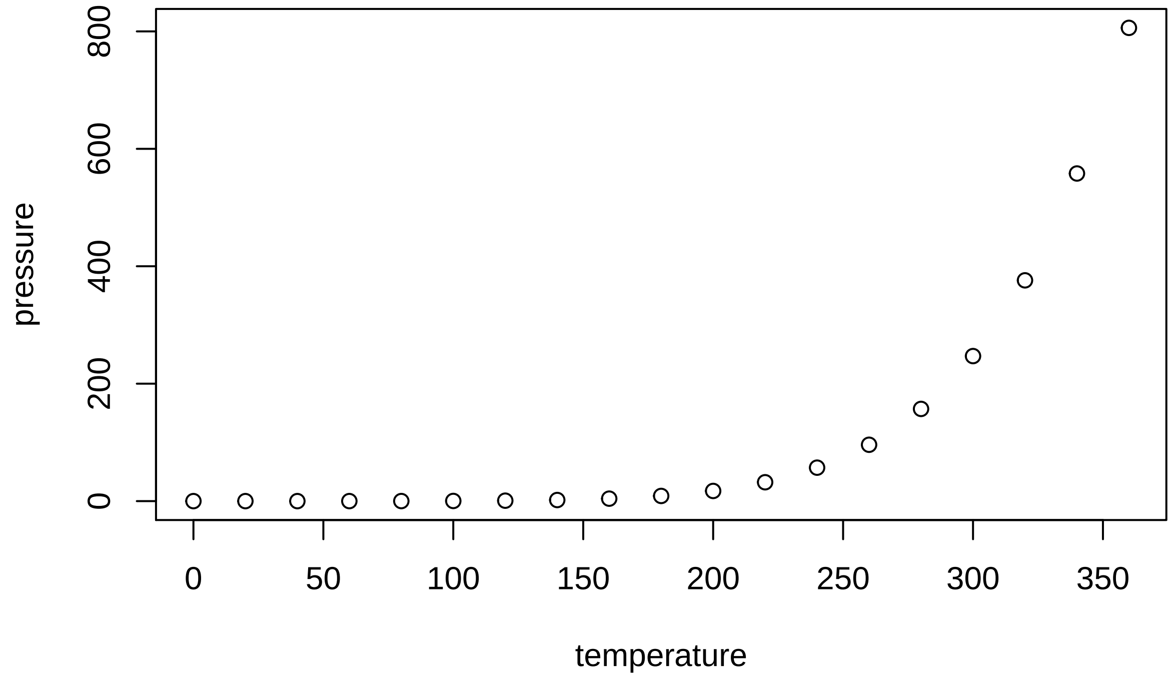
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

First I'm going to start by turning an SPSS file (i.e., .sav) into an RMarkdown file (i.e., .Rda). The data I will be using is from the World Bank. For more information on how to download World Bank data, please visit https://data.worldbank.org

Once the data is downloaded to a .SAV file, it is easy to use it using R by using the 'haven' package.

```
setwd("C:\\Users\\Juan Nunez\\Desktop\\MC_DATA_101\\ASSIG_1_DATA101")
##install.packages("haven")
library(haven)
```

Now I turn the .SAV file that is saved in my path into a .Rda file.

```
ASSIG1_DATA <- read_spss("C:\\Users\\Juan Nunez\\Desktop\\MC_DATA_101\\ASSIG_1_DATA101\\SPSS_DATA_FOR_R
```

Once the data set ASSIG1_DATA is in the environment, I can save it as an .Rda file.

```
save(ASSIG1_DATA,file="ASSIG1_DATA.Rda")
```

Now I can look at the data using 'dplyr'. First I download at bring up the package

```
##install.packages("dplyr")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
```

```
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Now I look at the dimensions of the ASSIG1_DATA data frame.

```r
dim(ASSIG1_DATA)
```

```
## [1] 148 356
```

I see that there are 148 rows and 356 columns. Let's look at the top 6 rows of this data frame.

```r
head(ASSIG1_DATA)
```

```
## # A tibble: 6 x 356
##      V1 V2    V3                  V4    V5    V6    V7        V8         V9
##   <dbl> <chr> <chr>            <dbl> <dbl> <dbl> <dbl>     <dbl>      <dbl>
## 1    1. AFGN  Afghanistan 1.83e-317   86.   NA    NA  4.77e-312  4.67e- 62
## 2    2. ALBN  Albania     3.50e+  1    NA   NA    NA  4.77e-312  4.67e- 62
## 3    3. ALGR  Algeria     5.30e+  1   55.   42.   36. 1.07e-314  4.15e-317
## 4    4. ANGL  Angola      2.80e+  1   72.   29.   NA  6.72e-318  4.67e- 62
## 5    5. ARGN  Argentina   8.70e+  1    5.   NA    NA  1.07e-314 -1.54e-180
## 6    6. ARMN  Armenia     6.80e+  1    NA   NA    NA  4.77e-312 -6.07e+ 66
## # ... with 347 more variables: V10 <dbl>, V11 <dbl>, V12 <dbl>, V13 <dbl>,
## #   V14 <dbl>, V15 <dbl>, V16 <dbl>, V17 <dbl>, V18 <dbl>, V19 <dbl>,
## #   V20 <dbl>, V21 <dbl>, V22 <dbl>, V23 <dbl>, V24 <dbl>, V25 <dbl>,
## #   V26 <dbl>, V27 <dbl>, V28 <dbl>, V29 <dbl>, V30 <dbl>, V31 <dbl>,
## #   V32 <dbl>, V33 <dbl>, V34 <dbl>, V35 <dbl>, V36 <dbl>, V37 <dbl>,
## #   V38 <dbl>, V39 <dbl>, V40 <dbl>, V41 <dbl>, V42 <dbl>, V43 <dbl>,
## #   V44 <dbl>, V45 <dbl>, V46 <dbl>, V47 <dbl>, V48 <dbl>, V49 <dbl>,
## #   V50 <dbl>, V51 <dbl>, V52 <dbl>, V53 <dbl>, V54 <dbl>, V55 <dbl>,
## #   V56 <dbl>, V57 <dbl>, V58 <dbl>, V59 <dbl>, V60 <dbl>, V61 <dbl>,
## #   V62 <dbl>, V63 <dbl>, V64 <dbl>, V65 <dbl>, V66 <dbl>, V67 <dbl>,
## #   V68 <dbl>, V69 <dbl>, V70 <dbl>, V71 <dbl>, V72 <dbl>, V73 <dbl>,
## #   V74 <dbl>, V75 <dbl>, V76 <dbl>, V77 <dbl>, V78 <dbl>, V79 <dbl>,
## #   V80 <dbl>, V81 <dbl>, V82 <dbl>, V83 <dbl>, V84 <dbl>, V85 <dbl>,
## #   V86 <dbl>, V87 <dbl>, V88 <dbl>, V89 <dbl>, V90 <dbl>, V91 <dbl>,
## #   V92 <dbl>, V93 <dbl>, V94 <dbl>, V95 <dbl>, V96 <dbl>, V97 <dbl>,
## #   V98 <dbl>, V99 <dbl>, V100 <dbl>, V101 <dbl>, V102 <dbl>, V103 <dbl>,
## #   V104 <dbl>, V105 <dbl>, V106 <dbl>, V107 <dbl>, V108 <dbl>,
## #   V109 <dbl>, ...
```

I see that the countries at the top of this data frame are Afghanistan, Albania, Algeria, Angola, Argentina, and Armenia. This data frame has way too many variables (i.e., columns) so I have to take a subset of the variables that I want to use. To take a subset of the data frame, I use the function 'select()'. The variables I am keeping are as coded as follows:

V1 COUNTRY NUMBER ; V2 ABBREVIATED COUNTRY NAME ; V3 COUNTRY NAME ; V5 % ADULT FEMALE ILLITERACY 1990 ; V12 ENERGY CONSUMPTION/CAPITA 1991 ; V14 INFANT MORTALITY RATE 1991 ; V168 F SCDRY SCH ENROL GER 1980 ; V133 CIVIL LIBERTIES 1991 ;

```r
NEW_ASSIG1_DATA2 <- select(ASSIG1_DATA, V1, V2, V3, V5, V12, V14, V168, V133)
```

Let's see what the top and bottom of this data frame looks like now.

```r
head(NEW_ASSIG1_DATA2)
```

```
## # A tibble: 6 x 8
##      V1 V2    V3             V5       V12       V14      V168  V133
##   <dbl> <chr> <chr>       <dbl>     <dbl>     <dbl>     <dbl> <dbl>
## 1    1. AFGN  Afghanistan   86. 9.00e+  1 1.83e-317 4.00e+  0    7.
## 2    2. ALBN  Albania       NA  1.85e-319 2.80e+  1 6.30e+  1    6.
## 3    3. ALGR  Algeria       55. 4.68e-317 6.40e+  1 2.60e+  1    4.
## 4    4. ANGL  Angola        72. 3.12e-317 1.30e+  2 9.00e+  0    7.
## 5    5. ARGN  Argentina      5. 4.68e-317 2.50e+  1 6.20e+  1    3.
## 6    6. ARMN  Armenia       NA  1.07e-314 2.20e+  1 1.83e-317   NA
```

```
tail(NEW_ASSIG1_DATA2)
```

```
## # A tibble: 6 x 8
##          V1 V2    V3                V5       V12       V14      V168  V133
##       <dbl> <chr> <chr>          <dbl>     <dbl>     <dbl>     <dbl> <dbl>
## 1 1.43e+  2 ZIMB  Zimbabwe         40. 1.31e-317 4.80e+  1 1.20e+  1    4.
## 2 1.31e-317 USSR  Soviet Union     NA  1.07e-314 1.83e-317 1.83e-317    4.
## 3 1.57e-317 FRG   Germany, West~   NA  4.75e-318 7.00e+  0 9.20e+  1   NA
## 4 1.83e-317 GDR   Germany, East~   NA  1.07e-314 1.83e-317 7.90e+  1   NA
## 5 2.09e-317 YMNA  Yemen ( Arab ~   NA  1.07e-314 1.83e-317 1.00e+  0   NA
## 6 2.35e-317 YMND  Yemen (PDR)      NA  1.07e-314 1.83e-317 1.10e+  1   NA
```

We still have 148 rows but now only 10 columns are left. Let's look at the descriptive statistics for V14.

```
summary(NEW_ASSIG1_DATA2$V14)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    0.00   14.00   35.50   48.98   83.00  149.00
```

Does V14 have any missing values?

```
is.na(NEW_ASSIG1_DATA2$V14)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [23] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [34] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [45] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [89] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [100] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [111] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [122] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [144] FALSE FALSE FALSE FALSE FALSE
```

It appears all cases are complete for V14, what about for V5?

```
is.na(NEW_ASSIG1_DATA2$V5)
```

```
##   [1] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE
##  [12] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [23] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
##  [34] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
##  [45] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE
##  [56] FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
##  [67] FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
##  [78]  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE
##  [89]  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
## [100] FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE
## [111] FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [122]  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE
## [133] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [144]  TRUE  TRUE  TRUE  TRUE  TRUE
```

We see that there are a number of cases that are missing for V5. So we are going to remove the missing cases from not only V5, but the rest of the data frame as well. In statistics, this methods of dealing with missing data is called listiwse deletion.

```r
ASSIG1FINAL <- complete.cases(NEW_ASSIG1_DATA2)
head(NEW_ASSIG1_DATA2[ASSIG1FINAL,])
```

```
## # A tibble: 6 x 8
##      V1 V2    V3             V5       V12       V14  V168  V133
##   <dbl> <chr> <chr>       <dbl>     <dbl>     <dbl> <dbl> <dbl>
## 1    1. AFGN  Afghanistan   86. 9.00e+  1 1.83e-317    4.    7.
## 2    3. ALGR  Algeria       55. 4.68e-317 6.40e+  1   26.    4.
## 3    4. ANGL  Angola        72. 3.12e-317 1.30e+  2    9.    7.
## 4    5. ARGN  Argentina      5. 4.68e-317 2.50e+  1   62.    3.
## 5    7. AUSL  Australia      2. 2.97e-317 8.00e+  0   72.    1.
## 6    8. AUST  Austria        2. 2.87e-317 8.00e+  0   87.    1.
```

The top of the data set doesn’t have any missing values, but we have to be sure.

```r
is.na(NEW_ASSIG1_DATA2[ASSIG1FINAL,])
```

```
##           V1    V2    V3    V5   V12   V14  V168  V133
##   [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [2,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [3,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [4,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [5,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [6,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [7,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [8,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   [9,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [10,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [11,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [12,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [14,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [15,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [16,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [17,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [18,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [19,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [20,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [21,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [22,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [23,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [24,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [26,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [27,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [28,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [29,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [30,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [31,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [32,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [33,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [34,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [35,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [36,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [38,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [39,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [40,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [41,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [42,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [43,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [44,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [45,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [46,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [47,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [48,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [50,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [51,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [52,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [53,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [54,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [55,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [56,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [57,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [58,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [59,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [60,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [62,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [63,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [64,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [65,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [66,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [67,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [68,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [69,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [70,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [71,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [72,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [74,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [75,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [76,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [77,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [78,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [79,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [80,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [81,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [82,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [83,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [84,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [86,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [87,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [88,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [89,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [90,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [91,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [92,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [93,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [94,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [95,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [96,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [98,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [99,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [100,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [101,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Maybe I don’t want to use V5 at all. How do I delete a column? I use the dplyr function select again.

```r
ASSIG1DATA3<-select(NEW_ASSIG1_DATA2, -V5)
head(ASSIG1DATA3)
```

```
## # A tibble: 6 x 7
##      V1 V2    V3                V12       V14      V168  V133
##   <dbl> <chr> <chr>           <dbl>     <dbl>     <dbl> <dbl>
## 1    1. AFGN  Afghanistan 9.00e+  1 1.83e-317 4.00e+  0    7.
## 2    2. ALBN  Albania     1.85e-319 2.80e+  1 6.30e+  1    6.
## 3    3. ALGR  Algeria     4.68e-317 6.40e+  1 2.60e+  1    4.
## 4    4. ANGL  Angola      3.12e-317 1.30e+  2 9.00e+  0    7.
## 5    5. ARGN  Argentina   4.68e-317 2.50e+  1 6.20e+  1    3.
## 6    6. ARMN  Armenia     1.07e-314 2.20e+  1 1.83e-317   NA
```

V5 is no longer part of the variables in this new subset. What if I was interested in only the countries that have high infant mortality rate? I can use the filter function to get that subset of the data.

```r
HIMR <- filter(ASSIG1DATA3, V14 > 50)
dim(HIMR)
```

```
## [1] 60  7
```

```r
head(HIMR)
```

```
## # A tibble: 6 x 7
##      V1 V2    V3               V12   V14  V168  V133
##   <dbl> <chr> <chr>          <dbl> <dbl> <dbl> <dbl>
## 1    3. ALGR  Algeria    4.68e-317   64.   26.    4.
## 2    4. ANGL  Angola     3.12e-317  130.    9.    7.
## 3   10. BNGL  Bangladesh 5.70e+  1  103.    6.    5.
## 4   13. BNIN  Benin      4.60e+  1  111.    9.    4.
## 5   14. BTAN  Bhutan     1.50e+  1  132.    1.    5.
## 6   15. BOLV  Bolivia    3.12e-317   83.   31.    3.
```

```r
tail(HIMR)
```

```
## # A tibble: 6 x 7
##      V1 V2    V3                  V12   V14      V168  V133
##   <dbl> <chr> <chr>             <dbl> <dbl>     <dbl> <dbl>
## 1  128. TRKY  Turkey       2.35e-317   58. 2.40e+  1    4.
## 2  129. TKMT  Turkmenistan 1.07e-314   56. 1.83e-317   NA
## 3  130. UGND  Uganda       2.50e+  1  118. 3.00e+  0    5.
## 4  139. YMNR  Yemen        9.60e+  1  109. 1.83e-317    5.
## 5  141. ZAIR  Zaire        7.10e+  1   94. 1.30e+  1    6.
## 6  142. ZMBA  Zambia       5.33e-318  106. 1.10e+  1    5.
```

We can see that a lot of countries have an infant mortality rate that is above 50 per 1000 births. Now what if I want to arrange the data according to infant mortality rate? I can use the 'arrange()' function.

```r
HIMR <- arrange(HIMR, V14)
head(HIMR)
```

```
## # A tibble: 6 x 7
##      V1 V2    V3                      V12   V14      V168  V133
##   <dbl> <chr> <chr>                 <dbl> <dbl>     <dbl> <dbl>
## 1  102. PERU  Peru             1.57e-317   53. 5.50e+  1    4.
## 2   35. DMNR  Dominican Rep.   2.61e-317   54. 1.83e-317    3.
## 3  115. SAFR  South Africa     5.59e-317   54. 1.83e-317    4.
## 4  100. PPNG  Papua New Guinea 7.27e-317   55. 8.00e+  0    3.
## 5   93. NCRG  Nicaragua        6.23e-317   56. 4.50e+  1    3.
## 6  129. TKMT  Turkmenistan     1.07e-314   56. 1.83e-317   NA
```

```r
tail(HIMR)
```

```
## # A tibble: 6 x 7
##      V1 V2    V3                  V12   V14  V168  V133
##   <dbl> <chr> <chr>             <dbl> <dbl> <dbl> <dbl>
## 1   19. BKFS  Burkina Faso 1.70e+  1  133.    2.    5.
## 2   49. GNEA  Guinea       6.80e+  1  136.   10.    5.
## 3   74. LBRA  Liberia      1.05e-317  136.   12.    7.
## 4   78. MLWI  Malawi       4.10e+  1  143.    2.    6.
## 5  112. SRLE  Sierra Leone 7.50e+  1  145.    8.    5.
## 6   87. MZBQ  Mozambique   5.90e+  1  149.    3.    6.
```

Out of the countries with more than 50 infant deaths per 1000 live births, we see that Peru is the country with the lowest infant mortality rate and that Mozambique is the country with the highest infant mortality rate. If I wnted to arrange this data in descending order, I can use the code bellow.

```r
HIMR <- arrange(HIMR, desc(V14))
head(HIMR)
```

```
## # A tibble: 6 x 7
##      V1 V2    V3                  V12   V14  V168  V133
##   <dbl> <chr> <chr>             <dbl> <dbl> <dbl> <dbl>
## 1   87. MZBQ  Mozambique   5.90e+  1  149.    3.    6.
## 2  112. SRLE  Sierra Leone 7.50e+  1  145.    8.    5.
## 3   78. MLWI  Malawi       4.10e+  1  143.    2.    6.
## 4   49. GNEA  Guinea       6.80e+  1  136.   10.    5.
## 5   74. LBRA  Liberia      1.05e-317  136.   12.    7.
## 6   19. BKFS  Burkina Faso 1.70e+  1  133.    2.    5.
```

```
tail(HIMR)
```

```
## # A tibble: 6 x 7
##      V1 V2    V3                     V12   V14      V168  V133
##   <dbl> <chr> <chr>                <dbl> <dbl>     <dbl> <dbl>
## 1   93. NCRG  Nicaragua        6.23e-317   56. 4.50e+  1    3.
## 2  129. TKMT  Turkmenistan     1.07e-314   56. 1.83e-317   NA
## 3  100. PPNG  Papua New Guinea 7.27e-317   55. 8.00e+  0    3.
## 4   35. DMNR  Dominican Rep.   2.61e-317   54. 1.83e-317    3.
## 5  115. SAFR  South Africa     5.59e-317   54. 1.83e-317    4.
## 6  102. PERU  Peru             1.57e-317   53. 5.50e+  1    4.
```

Everything looks good except for the variable names. So let’s change them using the ‘rename()’ function.

```
head(HIMR)
```

```
## # A tibble: 6 x 7
##      V1 V2    V3                 V12   V14  V168  V133
##   <dbl> <chr> <chr>            <dbl> <dbl> <dbl> <dbl>
## 1   87. MZBQ  Mozambique   5.90e+  1  149.    3.    6.
## 2  112. SRLE  Sierra Leone 7.50e+  1  145.    8.    5.
## 3   78. MLWI  Malawi       4.10e+  1  143.    2.    6.
## 4   49. GNEA  Guinea       6.80e+  1  136.   10.    5.
## 5   74. LBRA  Liberia      1.05e-317  136.   12.    7.
## 6   19. BKFS  Burkina Faso 1.70e+  1  133.    2.    5.
```

```
HIMR <- rename(HIMR, Country_ID = V1, Country_Code = V2, Country_Name = V3, Energy_Consumption_Per_Capi
head(HIMR)
```

```
## # A tibble: 6 x 7
##   Country_ID Country_Code Country_Name Energy_Consumptio~ Infant_Mortalit~
##        <dbl> <chr>        <chr>                     <dbl>            <dbl>
## 1        87. MZBQ         Mozambique             5.90e+  1             149.
## 2       112. SRLE         Sierra Leone           7.50e+  1             145.
## 3        78. MLWI         Malawi                 4.10e+  1             143.
## 4        49. GNEA         Guinea                 6.80e+  1             136.
## 5        74. LBRA         Liberia                1.05e-317             136.
## 6        19. BKFS         Burkina Faso           1.70e+  1             133.
## # ... with 2 more variables: Female_School_Enrollement <dbl>,
## #   CIVIL_LIBERTIES <dbl>
```

Sometime we want to transform variables in our data frame, we can use the funtion ‘mutate()’ to do that. Let’s remove the mean from V168.

```
HIMR <- mutate(HIMR, meanV168  = Female_School_Enrollement - mean(Female_School_Enrollement, na.rm = TRU
head(HIMR)
```

```
## # A tibble: 6 x 8
##   Country_ID Country_Code Country_Name Energy_Consumptio~ Infant_Mortalit~
##        <dbl> <chr>        <chr>                     <dbl>            <dbl>
## 1        87. MZBQ         Mozambique             5.90e+  1             149.
## 2       112. SRLE         Sierra Leone           7.50e+  1             145.
## 3        78. MLWI         Malawi                 4.10e+  1             143.
## 4        49. GNEA         Guinea                 6.80e+  1             136.
## 5        74. LBRA         Liberia                1.05e-317             136.
## 6        19. BKFS         Burkina Faso           1.70e+  1             133.
## # ... with 3 more variables: Female_School_Enrollement <dbl>,
## #   CIVIL_LIBERTIES <dbl>, meanV168 <dbl>
```

My new variable was added to the end of the data frame. Finally, we can use the 'group_by()' function to look at the descriptive statistics based on a criterion. In this example, we group data by infant mortality rate.

```
LIBERTIES <- group_by(HIMR, CIVIL_LIBERTIES)
head(LIBERTIES)
```

```
## # A tibble: 6 x 8
## # Groups:   CIVIL_LIBERTIES [3]
##   Country_ID Country_Code Country_Name Energy_Consumptio~ Infant_Mortalit~
##        <dbl> <chr>        <chr>                     <dbl>            <dbl>
## 1        87. MZBQ         Mozambique            5.90e+  1             149.
## 2       112. SRLE         Sierra Leone          7.50e+  1             145.
## 3        78. MLWI         Malawi                4.10e+  1             143.
## 4        49. GNEA         Guinea                6.80e+  1             136.
## 5        74. LBRA         Liberia               1.05e-317             136.
## 6        19. BKFS         Burkina Faso          1.70e+  1             133.
## # ... with 3 more variables: Female_School_Enrollement <dbl>,
## #   CIVIL_LIBERTIES <dbl>, meanV168 <dbl>
```

```
tail(LIBERTIES)
```

```
## # A tibble: 6 x 8
## # Groups:   CIVIL_LIBERTIES [3]
##   Country_ID Country_Code Country_Name  Energy_Consumpti~ Infant_Mortalit~
##        <dbl> <chr>        <chr>                     <dbl>            <dbl>
## 1        93. NCRG         Nicaragua             6.23e-317              56.
## 2       129. TKMT         Turkmenistan          1.07e-314              56.
## 3       100. PPNG         Papua New Gu~         7.27e-317              55.
## 4        35. DMNR         Dominican Re~         2.61e-317              54.
## 5       115. SAFR         South Africa          5.59e-317              54.
## 6       102. PERU         Peru                  1.57e-317              53.
## # ... with 3 more variables: Female_School_Enrollement <dbl>,
## #   CIVIL_LIBERTIES <dbl>, meanV168 <dbl>
```

Let's look at female school enrollment based on our new data frame.

```
summarize(LIBERTIES, Infant_Mortality_Rate = mean(Infant_Mortality_Rate, na.rm = TRUE))
```

```
## # A tibble: 6 x 2
##   CIVIL_LIBERTIES Infant_Mortality_Rate
##             <dbl>                 <dbl>
## 1              3.                  68.6
## 2              4.                  78.3
## 3              5.                 108.
## 4              6.                 107.
## 5              7.                 109.
## 6             NA                   56.0
```

We can see that the mean of the countries with the more infant mortality rates have less civil liberties (7 is lowest and 1 is the most liberties)