6_file_IO

Code ▾

input

1. `readline`

   1.syntax

   ```
   ```
   readline(prompt = "")
   ```
   return the string from the console
   ```

   1. ***!!!*** return is a string, needs to `as.integer()`
   2. eg

      ```
      # input and check for positive integer
      repeat{
              n <- as.numeric(readline(prompt="Enter the number of toys: "))
              if (is.na(n)) cat("Please enter a positive integer!")
              # check characters
              else if (n<=0) cat("Please enter a positive integer!")
              # check non-positive number
              else if (n-floor(n)>0) cat("Please enter a positive integer!")
              # check floats
              else break
          }
      ```

output

1. `print`

   Just give a var and print.

2. `sprintf`

   1. syntax: like the C printf, and the return is a string (in console, just print the return string on the console)
   2. detailed formats

      1. float

         ```
         > sprintf("Pi is %f", pi)
         # output real number with default option = 6 decimal places
         [1]  "Pi is 3.141593"
         > sprintf("%.3f", pi) # with 3 decimal places
         [1] "3.142"
         > sprintf("%5.1f", pi) # fixed width=5 with 1 decimal places
         [1] "  3.1"
         > sprintf("%-10f", pi) # left justified with fixed width=10
         [1] "3.141593 "
         > sprintf("%e", pi) # scientific notation
         [1] "3.141593e+00 "
         ```

      2. int: use `%d`

3. `cat`

   1. idea: just input multiple string, concatenate them together with auto coersion. No automatic space. Can also be used to print
   2. eg

      ```
      > cat("iteration = ", 7, "\n")
      iteration =  7

      # clear console
      cat('\f')
      ```

File I/O

1. misc configurations

1. working dir

   ```
   getwd()    # get
   setwd("C:/Folder")    # set
   ```

   alternatively, use IDE config …

1a. built-in data editer

R has a built-in data editor to help us to enter the data like using Excel.

If do `x<-edit(d1)` , the data editor window appears and we can edit the dataset `d1` , Close the window when finish and we will have our data stored in `x` .

2. input objects

1. `read.table`

   1. Idea: Read from file in `dat` format, or tsv … Need to be separated by tab (?) Return a dataframe.
   2. syntax

      ```
      read.table(file, header = FALSE, sep = "", quote = "\"'",
                 dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
                 row.names, col.names, as.is = !stringsAsFactors, tryLogical = TRUE,
                 na.strings = "NA", colClasses = NA, nrows = -1,
                 skip = 0, check.names = TRUE, fill = !blank.lines.skip,
                 strip.white = FALSE, blank.lines.skip = TRUE,
                 comment.char = "#",
                 allowEscapes = FALSE, flush = FALSE,
                 stringsAsFactors = FALSE,
                 fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
      ```

      1. `header` : if the files contains the header
   3. egs

      ```
      # read a typical csv
      data <- read.table("ex2_q3.dat", sep=",", header=T, stringsAsFactorfs=T)
      ```

2. `read.csv`

   ```
   read.csv(file, header = TRUE, sep = ",", quote = "\"",
            dec = ".", fill = TRUE, comment.char = "", ...)
   ```

3. Store objects

1. store images: see section 4
2. store data objects

   1. `write.table`
      1. idea: write to an ASCII file
      2. syntax

         ```
         write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ",
                     eol = "\n", na = "NA", dec = ".", row.names = TRUE,
                     col.names = TRUE, qmethod = c("escape", "double"),
                     fileEncoding = "")
         ```

         1. row.names / col.names: logical value: row / col names of x are to be written along with x, or a character vector of row / col names to be written.
      3. eg

         ```
         table(x,file="popden1.dat",row.names=F)
         # row.names=F is important. otherwise add in the row number in x.
         ```

   2. `write.csv`