R-Basic-Notes.3

Charlene Garridos

2023-03-31

Url of data set

```
url <- "https://raw.githubusercontent.com/msuiitdmsgabriel/datasets-regression/main/salespeople.csv"
```

#load the data set and store it as a dataframe called salespeople

```
salespeople <- read.csv(url)
local_salespeople <- read.csv("salespeople.csv")
```

#the function head, which displays thee first rows of a dataframe, has only one required argument x: the name of the dataframe

```
high_sales <- subset(salespeople, subset = sales >= 700)
head(high_sales)
```

```
##    promoted sales customer_rate performance
## 6         1   918          4.54           2
## 12        1   716          3.16           3
## 20        1   937          5.00           2
## 21        1   702          3.53           4
## 25        1   819          4.45           2
## 26        1   736          3.94           4
```

#To select specific columns use the select argument

```
subset(salespeople, select = c("sales", "performance"))
```

```
salespeople_sales_part <- subset(salespeople, select = c("sales", "performance"))
head(salespeople_sales_part)
```

```
##   sales performance
## 1   594           2
## 2   446           3
## 3   674           4
## 4   525           2
## 5   657           3
## 6   918           2
```

head(salespeople_sales_perf)

```
low_sales <- subset(salespeople, subset = sales < 400)
```

#binds the rows of low_sales and high_sales together

```
low_and_high_sales = rbind(low_sales, high_sales)
head(low_and_high_sales)
```

```
##    promoted sales customer_rate performance
## 7         0   318          3.09           3
## 8         0   364          4.89           1
## 9         0   342          3.74           3
## 10        0   387          3.00           3
## 15        0   344          3.02           2
## 16        0   372          3.87           3
```

```
head(high_sales)
```

```
##    promoted sales customer_rate performance
## 6         1   918          4.54           2
## 12        1   716          3.16           3
## 20        1   937          5.00           2
## 21        1   702          3.53           4
## 25        1   819          4.45           2
## 26        1   736          3.94           4
```

```
head(low_and_high_sales)
```

```
##    promoted sales customer_rate performance
## 7         0   318          3.09           3
## 8         0   364          4.89           1
## 9         0   342          3.74           3
## 10        0   387          3.00           3
## 15        0   344          3.02           2
## 16        0   372          3.87           3
```

#two dataframes with two columns each

```
sales_perf <- subset(salespeople, select = c("sales", "performance"))
prom_custrate <- subset(salespeople, select =  c("promoted", "customer_rate"))
```

#bind the columns to create a dataframes with four columns

```
full_df <- cbind(sales_perf, prom_custrate)
head(full_df)
```

```
##   sales performance promoted customer_rate
## 1   594           2        0          3.94
## 2   446           3        0          4.06
```

```
## 3   674           4        1          3.83
## 4   525           2        0          3.62
## 5   657           3        1          4.40
## 6   918           2        1          4.54
```

```
head(sales_perf)
```

```
##   sales performance
## 1   594           2
## 2   446           3
## 3   674           4
## 4   525           2
## 5   657           3
## 6   918           2
```

```
head(prom_custrate)
```

```
##   promoted customer_rate
## 1        0          3.94
## 2        0          4.06
## 3        1          3.83
## 4        0          3.62
## 5        1          4.40
## 6        1          4.54
```

```
head(full_df)
```

```
##   sales performance promoted customer_rate
## 1   594           2        0          3.94
## 2   446           3        0          4.06
## 3   674           4        1          3.83
## 4   525           2        0          3.62
## 5   657           3        1          4.40
## 6   918           2        1          4.54
```

```
head(full_df, n=5)
```

```
##   sales performance promoted customer_rate
## 1   594           2        0          3.94
## 2   446           3        0          4.06
## 3   674           4        1          3.83
## 4   525           2        0          3.62
## 5   657           3        1          4.40
```

Functions, Packages and Libraries

Functions are operations that take certain defined inputs and return an output. Functions exist to perform common useful operations. It usually take one or more arguments. Often there are a large number of arguments that a function can take, but many are optional and not required to be specified by the user. A

second argument is optional, n: the number of rows to display. If n is not entered, it is assumed to have the default value n = 6. When running a function, you can either specify the arguments by name or you can enter them in order without their names. If you enter without naming, R expects the arguments to be entered in exactly the right order.

```
head(x = salespeople)
```

```
##   promoted sales customer_rate performance
## 1        0   594          3.94           2
## 2        0   446          4.06           3
## 3        1   674          3.83           4
## 4        0   525          3.62           2
## 5        1   657          4.40           3
## 6        1   918          4.54           2
```

#see fewer - arguments need to be in the right order if not named

```
head(salespeople, 6)
```

```
##   promoted sales customer_rate performance
## 1        0   594          3.94           2
## 2        0   446          4.06           3
## 3        1   674          3.83           4
## 4        0   525          3.62           2
## 5        1   657          4.40           3
## 6        1   918          4.54           2
```

#or if you don't know the right order, name your arguments and you can put them in any order

```
head(x = salespeople, n = 4)
```

```
##   promoted sales customer_rate performance
## 1        0   594          3.94           2
## 2        0   446          4.06           3
## 3        1   674          3.83           4
## 4        0   525          3.62           2
```

```
head(n = 6, x = salespeople)
```

```
##   promoted sales customer_rate performance
## 1        0   594          3.94           2
## 2        0   446          4.06           3
## 3        1   674          3.83           4
## 4        0   525          3.62           2
## 5        1   657          4.40           3
## 6        1   918          4.54           2
```

To get help on the head() function,

Type help(head) or ?head which will display the results in the Help browser window in Rstudio. Alternatively you can open the Help browser window directly in Rstudio and do a search there. Users can write their own functions to perform tasks that are helpful to their objective. Functions are not limited to those that come packaged in R.

```
help()
```

```
## starting httpd help server ... done
```

```
help(head)
```

```
?head
```

Experienced programmers in most languages subscribe to a principle called DRY (Don't Repeat Yourself).Whenever a task needs to be done repeatedly, it is poor practice to write the same code numerous times.

```
df_report <- function(df){paste("This dataframe contains", nrow(df), "rows and", ncol(df), "columns. Th
```