An Overview of the R Language

Luis Francisco Gómez López

FAEDIS

2024-08-01

SCIENTIÆ · PATRIÆ · FAMILIÆ

UNIVERSIDAD MILITAR
NUEVA GRANADA

Table of contents I

- This presentation is based on (Chapman and Feit 2019, chap. 2)

- Equip beginners with a programming foundation by leveraging the R language, enabling practical application in marketing research and analytics

- **Ordinal 7 point scale**
  - Extremely satisfied: 7
  - Moderately satisfied: 6
  - Slightly satisfied: 5
  - Neither satisfied or dissatisfied: 4
  - Slightly dissatisfied: 3
  - Moderately dissatisfied: 2
  - Extremely dissatisfied: 1
- **Variables**
  - iProdSAT: satisfaction with a product
  - iSalesSAT: satisfaction with sales experience
  - iProdREC: likelihood to recommend the product
  - iSalesREC: likelihood to recommend the sales person
  - Segment: market segment assigned by a clustering algorithm (Chapman and Feit 2019, chap. 11)

- **Import data: the base R way**

```
satisfaction_data <- read.csv(file = "http://goo.gl/UDv12g")
satisfaction_data |> head(n=5)
```

```
  iProdSAT iSalesSAT Segment iProdREC iSalesREC
1        6         2       1        4         3
2        4         5       3        4         4
3        5         3       4        5         4
4        3         3       2        4         4
5        3         3       3        2         2
```

- **Import data: the tidyverse way**

```r
library(tidyverse) # Remember to load the tidyverse library
satisfaction_data <- read_csv(file = "http://goo.gl/UDv12g")
satisfaction_data |> head(n=5)
```

```
# A tibble: 5 x 5
  iProdSAT iSalesSAT Segment iProdREC iSalesREC
     <dbl>     <dbl>   <dbl>    <dbl>     <dbl>
1        6         2       1        4         3
2        4         5       3        4         4
3        5         3       4        5         4
4        3         3       2        4         4
5        3         3       3        2         2
```

- **Transform data: the base R way**

```r
satisfaction_data$Segment <- factor(x = satisfaction_data$Segment,
                                    ordered = FALSE)
summary(object = satisfaction_data)
```

```
    iProdSAT        iSalesSAT       Segment     iProdREC         iSalesREC
 Min.   :1.00    Min.   :1.000    1: 54    Min.   :1.000    Min.   :1.000
 1st Qu.:3.00    1st Qu.:3.000    2:131    1st Qu.:3.000    1st Qu.:3.000
 Median :4.00    Median :4.000    3:154    Median :4.000    Median :3.000
 Mean   :4.13    Mean   :3.802    4:161    Mean   :4.044    Mean   :3.444
 3rd Qu.:5.00    3rd Qu.:5.000             3rd Qu.:5.000    3rd Qu.:4.000
 Max.   :7.00    Max.   :7.000             Max.   :7.000    Max.   :7.000
```

- **Transform data: the skimr and the tidyverse way**
  - Ups the table is really big!!! Try it in your console to see the complete table

```
library(skimr) # Remember to install the package if it is not installed
satisfaction_data <- satisfaction_data |>
  mutate(Segment = factor(x = Segment, ordered = FALSE))
```

```
satisfaction_data |> skim()
```

- **R objects**: everything in R is an object (object-oriented). For now, we will only inspect a few selected objects:
  - **Atomic vectors**[1]
    - Logical
    - Integer
    - Double
  - **Factors**
  - **Data Frames**
  - **Tibbles**

[1] In R the atomic vectors are logical, integer, double, numeric (which includes integer and double), character, complex and raw but for pedagogical purposes we are going to check later character, numeric includes integer and double and we are not going to use complex and raw

- **Atomic vectors**
  - Logical

```r
as.integer(satisfaction_data$Segment)[1:5] == 1
```

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

```r
as.integer(satisfaction_data$Segment)[1:5] > 1
```

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

```r
as.integer(satisfaction_data$Segment)[1:5] >= 1
```

```
[1] TRUE TRUE TRUE TRUE TRUE
```

- **Atomic vectors**
  - Integer

```r
as.integer(satisfaction_data$Segment)[1:5]
```

```
[1] 1 3 4 2 3
```

- **Atomic vectors**
  - Double

```r
sprintf("%.2f", satisfaction_data$iProdSAT[1:5])
```

```
[1] "6.00" "4.00" "5.00" "3.00" "3.00"
```

- **Factors**

```
satisfaction_data$Segment[1:5]
```

```
[1] 1 3 4 2 3
Levels: 1 2 3 4
```

- **Data Frames**

```
as.data.frame(satisfaction_data) |> head(n=5)
```

```
  iProdSAT iSalesSAT Segment iProdREC iSalesREC
1        6         2       1        4         3
2        4         5       3        4         4
3        5         3       4        5         4
4        3         3       2        4         4
5        3         3       3        2         2
```

- **Tibble**

```
class(satisfaction_data)
```

```
[1] "tbl_df"     "tbl"        "data.frame"
```

```
satisfaction_data |> head(n=5)
```

```
# A tibble: 5 x 5
  iProdSAT iSalesSAT Segment iProdREC iSalesREC
     <dbl>     <dbl> <fct>      <dbl>     <dbl>
1        6         2 1              4         3
2        4         5 3              4         4
3        5         3 4              5         4
4        3         3 2              4         4
5        3         3 3              2         2
```

- **Add new variables: the base R way**

```
satisfaction_data$customer <- 1:nrow(satisfaction_data)
as.data.frame(satisfaction_data) |> head(n=5)
```

```
  iProdSAT iSalesSAT Segment iProdREC iSalesREC customer
1        6         2       1        4         3        1
2        4         5       3        4         4        2
3        5         3       4        5         4        3
4        3         3       2        4         4        4
5        3         3       3        2         2        5
```

- **Add new variables: the tidyverse way**

```
satisfaction_data |>
  mutate(customer = 1:nrow(satisfaction_data)) |>
  head(n=5)
```

```
# A tibble: 5 x 6
  iProdSAT iSalesSAT Segment iProdREC iSalesREC customer
     <dbl>     <dbl> <fct>      <dbl>     <dbl>    <int>
1        6         2 1              4         3        1
2        4         5 3              4         4        2
3        5         3 4              5         4        3
4        3         3 2              4         4        4
5        3         3 3              2         2        5
```

- **Picks variables based on their names: the base R way**

```
as.data.frame(satisfaction_data)[c("customer", "Segment",
                                   "iProdSAT", "iSalesSAT", "iProdREC", "iSalesREC")] |>
  head(n=5)
```

```
  customer Segment iProdSAT iSalesSAT iProdREC iSalesREC
1        1       1        6         2        4         3
2        2       3        4         5        4         4
3        3       4        5         3        5         4
4        4       2        3         3        4         4
5        5       3        3         3        2         2
```

- **Picks variables based on their names: the tidyverse way**

```
satisfaction_data |>
  select(customer, Segment, iProdSAT, iSalesSAT, iProdREC, iSalesREC) |>
  head(n=5)
```

```
# A tibble: 5 x 6
  customer Segment iProdSAT iSalesSAT iProdREC iSalesREC
     <int> <fct>      <dbl>     <dbl>    <dbl>     <dbl>
1        1 1              6         2        4         3
2        2 3              4         5        4         4
3        3 4              5         3        5         4
4        4 2              3         3        4         4
5        5 3              3         3        2         2
```

- **Picks cases based on their values: the base R way**

```r
as.data.frame(satisfaction_data)[satisfaction_data$Segment == 2, ] |>
  head(n=5)
```

```
   iProdSAT iSalesSAT Segment iProdREC iSalesREC customer
4         3         3       2        4         4        4
14        4         3       2        3         2       14
18        3         5       2        3         3       18
19        4         4       2        1         1       19
23        4         2       2        4         6       23
```

- **Picks cases based on their values: the tidyverse way**

```r
satisfaction_data |>
  filter(Segment == 2) |>
  head(n=5)
```

```
# A tibble: 5 x 6
  iProdSAT iSalesSAT Segment iProdREC iSalesREC customer
     <dbl>     <dbl> <fct>      <dbl>     <dbl>    <int>
1        3         3 2              4         4        4
2        4         3 2              3         2       14
3        3         5 2              3         3       18
4        4         4 2              1         1       19
5        4         2 2              4         6       23
```

- **Reduces multiple values to a single summary: the base R way**

```r
data.frame(mean_iProdSAT = mean(satisfaction_data$iProdSAT),
           median_iSalesSAT = median(satisfaction_data$iSalesSAT))
```

```
  mean_iProdSAT median_iSalesSAT
1          4.13                4
```

- **Reduces multiple values to a single summary: the tidyverse way**

```r
satisfaction_data |>
  summarise(mean_iProdSAT = mean(iProdSAT), median_iSalesSAT = median(iSalesSAT))
```

```
# A tibble: 1 x 2
  mean_iProdSAT median_iSalesSAT
          <dbl>            <dbl>
1          4.13                4
```

- **Does product and sales satisfaction differ by segment?: the base R way**

```
satisfaction_data[c("iProdSAT", "iSalesSAT")] |>
  aggregate(by = satisfaction_data[c("Segment")], FUN = mean) |>
  setNames(nm = c("Segment", "mean_iProdSAT", "mean_iSalesSAT"))
```

```
  Segment mean_iProdSAT mean_iSalesSAT
1       1      3.462963       2.981481
2       2      3.725191       3.381679
3       3      4.103896       3.811688
4       4      4.708075       4.409938
```

- **Does product and sales satisfaction differ by segment?: the tidyverse way**

```
satisfaction_data |>
  group_by(Segment) |>
  select(iProdSAT, iSalesSAT) |>
  summarise(mean_iProdSAT = mean(iProdSAT), mean_iSalesSAT = mean(iSalesSAT))
```

```
# A tibble: 4 x 3
  Segment mean_iProdSAT mean_iSalesSAT
  <fct>           <dbl>          <dbl>
1 1                3.46           2.98
2 2                3.73           3.38
3 3                4.10           3.81
4 4                4.71           4.41
```

- To my family that supports me
- To the taxpayers of Colombia and the **UMNG students** who pay my salary
- To the **Business Science** and **R4DS Online Learning** communities where I learn **R** and **π-thon**
- To the **R Core Team**, the creators of **RStudio IDE**, **Quarto** and the authors and maintainers of the packages **tidyverse**, **skimr** and **tinytex** for allowing me to access these tools without paying for a license
- To the **Linux kernel community** for allowing me the possibility to use some **Linux distributions** as my main **OS** without paying for a license

References I

Chapman, Chris, and Elea McDonnell Feit. 2019. *R For Marketing Research and Analytics*. 2nd ed. 2019. Use R! Cham: Springer International Publishing : Imprint: Springer. https://doi-org.ezproxy.umng.edu.co/10.1007/978-3-030-14316-9.