R Commands

Faith Kabanda

2022-07-25

## 1. For Loops

for(i in c(-3, 5, 4, 7, 9, 4, 5)){  
 + print(i^2)  
}

## [1] 9  
## [1] 25  
## [1] 16  
## [1] 49  
## [1] 81  
## [1] 16  
## [1] 25

## 2. Functions

A function is a set of statements organized together to perform a specific task. R has a large number of in-built functions and the user can create their own functions.

*function\_name <- function(arg\_1, arg\_2, …) {*

*Function body*

*}*

# Create a sequence of numbers from 32 to 44.  
print(seq(32,44))

## [1] 32 33 34 35 36 37 38 39 40 41 42 43 44

# Find mean of numbers from 25 to 82.  
print(mean(25:82))

## [1] 53.5

# Find sum of numbers frm 41 to 68.  
print(sum(41:68))

## [1] 1526

## Group By And %>%

Group\_by() function belongs to the dplyr package in the R programming language, which groups the data frames. Group\_by() function alone will not give any output. It should be followed by summarise() function with an appropriate action to perform. It works similar to GROUP BY in SQL and pivot table in excel.

%>% is called the forward pipe operator in R. It provides a mechanism for chaining commands with a new forward-pipe operator, %>%. This operator will forward a value, or the result of an expression, into the next function call/expression

library(readxl)  
library(magrittr) #to use the pipe command

## Warning: package 'magrittr' was built under R version 4.2.1

library(dplyr) #for the summarise function to work

##   
## 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

candy\_data<- read\_excel("C:/Users/Faith Kabanda/OneDrive/Documents/L-IFT/GitHub Test folder/testfolder/testfolder/Sweet Tooth Distribution Week 1 Trainees.xlsx",   
 sheet = "Distribution Records Bulawayo")  
candy\_data

## # A tibble: 59 × 8  
## `Delivery Date` `Receipt Number` Branch Category Product Quantity  
## <dttm> <dbl> <chr> <chr> <chr> <dbl>  
## 1 2021-02-04 00:00:00 61803 Bulawayo Cookies Arrowroot 58  
## 2 2021-02-25 00:00:00 61810 Bulawayo Cookies Arrowroot 30  
## 3 2021-05-01 00:00:00 61832 Bulawayo Cookies Arrowroot 77  
## 4 2021-07-27 00:00:00 61861 Bulawayo Cookies Arrowroot 20  
## 5 2020-02-24 00:00:00 61688 Bulawayo Bars Bran 42  
## 6 2021-01-11 00:00:00 61795 Bulawayo Bars Bran 77  
## 7 2021-07-03 00:00:00 61853 Bulawayo Bars Bran 65  
## 8 2021-09-07 00:00:00 61875 Bulawayo Bars Bran 50  
## 9 2021-10-01 00:00:00 61883 Bulawayo Bars Bran 43  
## 10 2021-12-03 00:00:00 61904 Bulawayo Bars Bran 42  
## # … with 49 more rows, and 2 more variables: UnitPrice <dbl>, TotalPrice <dbl>

candy\_data\_by\_category<- candy\_data %>% group\_by(Category) %>% summarise(SubTotal= sum(TotalPrice))  
candy\_data\_by\_category

## # A tibble: 5 × 2  
## Category SubTotal  
## <chr> <dbl>  
## 1 Bars 2886.  
## 2 Cookies 4138.  
## 3 Crackers 147.  
## 4 Snacks 659.  
## 5 <NA> NA

## Mutate

What is mutate in R? In R programming, the mutate function is used to create a new variable from a data set. In order to use the function, we need to install the *dplyr* package.

candy\_data\_mutate<- candy\_data %>% mutate(total\_production\_cost = TotalPrice\*0.85)  
candy\_data\_mutate

## # A tibble: 59 × 9  
## `Delivery Date` `Receipt Number` Branch Category Product Quantity  
## <dttm> <dbl> <chr> <chr> <chr> <dbl>  
## 1 2021-02-04 00:00:00 61803 Bulawayo Cookies Arrowroot 58  
## 2 2021-02-25 00:00:00 61810 Bulawayo Cookies Arrowroot 30  
## 3 2021-05-01 00:00:00 61832 Bulawayo Cookies Arrowroot 77  
## 4 2021-07-27 00:00:00 61861 Bulawayo Cookies Arrowroot 20  
## 5 2020-02-24 00:00:00 61688 Bulawayo Bars Bran 42  
## 6 2021-01-11 00:00:00 61795 Bulawayo Bars Bran 77  
## 7 2021-07-03 00:00:00 61853 Bulawayo Bars Bran 65  
## 8 2021-09-07 00:00:00 61875 Bulawayo Bars Bran 50  
## 9 2021-10-01 00:00:00 61883 Bulawayo Bars Bran 43  
## 10 2021-12-03 00:00:00 61904 Bulawayo Bars Bran 42  
## # … with 49 more rows, and 3 more variables: UnitPrice <dbl>, TotalPrice <dbl>,  
## # total\_production\_cost <dbl>

## Vectors

Vectors are the simplest data structures in R. They are sequences of elements of the same basic type. These types can be numeric, integer, complex, character, and logical. In R, the more complicated data structures are made with vectors as building-blocks.

To create a vector, we use the c() function.