second Task

Alejandro Vergara Rincon 81190

2023-08-29

1

After making the pertinent connections with Github, there will be the DESCRIPTION file and we proceed to fill in the required spaces, in this case it is filled like this:

Package: test1

Type: Package

Title: Package for task 2 development.

Version: 0.1.0

Author: Jose Alejandro Vergara 81190 Maintainer:

The package maintainer Josea.vergararar@ecci.edu.co

Description: This code is used to solve the data transformation exercises:https://r4ds.had.co.nz/transform.html#transform

License: "use_mit_license"

Encoding: UTF-8

LazyData: true

Imports: dplyr, tidyr, knitr

RoxygenNote: 7.2.3

2

```r
SYA <- function(x) {
  library(nycflights13)
  library(tidyverse)
  library(knitr)
  if (x == 1) {
    return(Ex1())
  } else if (x == 2) {
    return(Ex2())
  } else if (x == 3) {
    return(Ex3())}
  else if (x == 4) {
  return(Ex4())}
  else if (x == 5) {
    return(Ex5())}
```

```
  else if (x == 6) {
    return(Ex6())}
  else {

    print("It is not possible to perform this operation")
  }
}
```

This code defines an R function named **SYA** (search_your_asnwer). This function takes a single argument **x**, which is used to determine which specific task or operation to perform. Let's break down the code step by step:

1. **Loading Libraries**: The function starts by loading several R packages: **nycflights13**, **tidyverse**, and **knitr**. These packages provide various functions and tools for data manipulation, analysis, and visualization.

2. **Conditional Execution**: The function uses a series of **if** and **else if** statements to determine what action to take based on the value of the input **x**.

   - If **x** is equal to 1, the function will return the result of the **Ex1()** function.
   - If **x** is equal to 2, the function will return the result of the **Ex2()** function.
   - If **x** is equal to 3, the function will return the result of the **Ex3()** function.
   - If **x** is equal to 4, the function will return the result of the **Ex4()** function.
   - If **x** is equal to 5, the function will return the result of the **Ex5()** function.
   - If **x** is equal to 6, the function will return the result of the **Ex6()** function.
   - If none of the above conditions are met, a message "It is not possible to perform this operation" will be printed.

3. **Function Execution**: The function will execute the specified sub-function (e.g., **Ex1()**, **Ex2()**, etc.) based on the value of **x** that was passed as an argument. Each **Ex1**, **Ex2, etc.** corresponds to applying the solutions developed in the previous task, therefore the solution code for each one is identical to the solution in the first assignment.

Example

Here it is evident how by putting SYA(1) it extracts what was done in exercise 1 with a short explanation

```
library("test1")
SYA(1)
```

```
## -- Attaching core tidyverse packages ---------------------- tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
##
## 5.2.4 Exercises:
## 1
## #The first step is to load the nycflights13 and tidyverse libraries:
##
##
## library(nycflights13)
## library(tidyverse)
## library(knitr)
##
##
## #Then the flight data is assigned to a called variable,
##  prin this case the variable exercise This allows you to use the variable exercise to reference
##  the data in the code:
##
##
## exercise <- nycflights13::flights
##
##
## Data_1 <- filter(exercise, arr_delay >= 2)
##
##
## #A new dataset named Data_1 is created by using the filter() function,
## which is part of the dplyr package within tidyverse. This function allows you to filter
## the data according to a given condition.
##
##
##
##
## Table: In this table, you can see information about the flights.
##
## | year | dest | origin | arr_delay |
## |:----:|:----:|:------:|:---------:|
## | 2013 | IAH  | LGA    |    20     |
## | 2013 | MIA  | JFK    |    33     |
## | 2013 | ORD  | LGA    |    8      |
## | 2013 | LAX  | JFK    |    7      |
## | 2013 | DFW  | LGA    |    31     |
## | 2013 | ORD  | EWR    |    32     |
## | 2013 | RSW  | JFK    |    4      |
## | 2013 | PHX  | EWR    |    3      |
## | 2013 | MIA  | LGA    |    5      |
## | 2013 | MSP  | EWR    |    29     |
##
## 2
## Data_2 <- filter(exercise, dest == "IAH" | dest == "HOU")
##
## # In this code, we use the OR operator (\|) to filter out flights that are bound for Houston,
## either IAH or HOU. The condition dest == "IAH" \| dest == "HOU" ensures that flights that meet
## either of these two options are included.
##
##
##
##
```

Table: In this table, you can see information about the flights and destinations.
##
| year | dest | origin | arr_delay |
|:----:|:----:|:------:|:---------:|
| 2013 | IAH | EWR | 11 |
| 2013 | IAH | LGA | 20 |
| 2013 | IAH | LGA | 1 |
| 2013 | IAH | LGA | 3 |
| 2013 | IAH | EWR | 26 |
| 2013 | IAH | EWR | 9 |
| 2013 | IAH | LGA | 11 |
| 2013 | IAH | EWR | 1 |
| 2013 | IAH | LGA | 145 |
| 2013 | IAH | EWR | -2 |
| 2013 | HOU | JFK | 38 |
| 2013 | IAH | LGA | 0 |
| 2013 | HOU | EWR | 12 |
| 2013 | IAH | EWR | 19 |
| 2013 | IAH | LGA | -9 |