Tidyverse Create Recipe

Folorunsho Atanda

2023-11-05

For this assignment I decided to use the **map()** function in the **purrr* package of the tidyverse.

I chose **map()** because it allows for element operation of a vector or list. It gives us a replacement for **for loops**.

```
library(tidyverse)
```

```
## -- 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   3.4.3     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
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
url <- "https://raw.githubusercontent.com/folushoa/Data-Science/Data-607/Tidyverse/heart_failure_clinica

data <- fread(url)
```

```r
#this function take a value and checks if it is **0** or **1**. By default if the value is **0** it ret
binary_fn <- function(x, true_value = "No", false_value = "Yes"){
  result <- if_else(x == 0, true_value, false_value)
  return(result)
}
```

```r
#using the function, replace the values of some of the columns
data$sex <- map(data$sex, ~binary_fn(.x, "Female", "Male"))

data$high_blood_pressure <- map(data$high_blood_pressure, ~binary_fn(.x))

data$anaemia <- map(data$anaemia, ~binary_fn(.x))

data$diabetes <- map(data$diabetes, ~binary_fn(.x))

data$smoking <- map(data$smoking, ~binary_fn(.x,))

data$DEATH_EVENT <- map(data$DEATH_EVENT, ~binary_fn(.x))
```

For each of these columns, **map()** applied **binary__fn** to each of the elements of the column.

This allows for better understanding of our data. Take for example

```r
#count of observations grouped by Death Event
data_by_death_event <- data %>%
  group_by(DEATH_EVENT) %>%
  count()
print(data_by_death_event)
```

```
## # A tibble: 2 x 2
## # Groups:   DEATH_EVENT [2]
##   DEATH_EVENT     n
##   <list>      <int>
## 1 <chr [1]>      96
## 2 <chr [1]>     203
```

```r
#count of observations grouped by sex and Death Event
data_by_sex_n_death_event <- data %>%
  group_by(sex, DEATH_EVENT) %>%
  count()
print(data_by_sex_n_death_event)
```

```
## # A tibble: 4 x 3
## # Groups:   sex, DEATH_EVENT [4]
##   sex       DEATH_EVENT     n
##   <list>    <list>      <int>
## 1 <chr [1]> <chr [1]>      62
## 2 <chr [1]> <chr [1]>     132
## 3 <chr [1]> <chr [1]>      34
## 4 <chr [1]> <chr [1]>      71
```

Note: I don't understand why my columns have the value **<chr [1]>**. When I view the data frame the values are **"Yes", "No", "Male", or "Female"**.