R source code for loading LOS_model dataset from NHSRdatasets

B213753

15/06/2022

Link to github repo

https://github.com/B213753/B213753_assessment.git

Introduction

The data capture tool will look at the (length of stay) 'LOS_model' dataset and attempt to model hospital length of stay (LOS) in the ten different trusts within the dataset. Being able to analyse the differences, if any, in length of stay in different trusts will allow for better planning and allocation of resources, or perhaps highlight areas for improvement with future more in-depth analyses.

The tool will look to collect the following variables: patient ID, organisation, patient age, length of stay, and death. Please see the data dictionary below for more details on these variables.

Script function and output

The purpose of this script is to load the LOS_model dataset into R, give a brief overview of the data using summary tables, recode the death variable into a factor and finally split the dataset into test and train sets.

The script also uses R to build a data dictionary that will be appended onto the data collected with the data capture tool.

The formal data dictionary is also provided in this document.

Outputs

The following files will be created/modified on successful conclusion of this script. This script assumes the working directory is set as the root project directory.

- */RawData/LOS_raw.csv* - The raw LOS data containing 300 observations.
- */Data/LOS_train.csv* - The train dataset containing 288 observations.
- */Data/LOS_test.csv* - The test dataset containing 11 observations.
- */Data/LOS_test_marker.csv* - The test marker dataset containing 1 observation.

Add the dictionary stuff here !!!!! !!!!!

Load packages and data

The following packages are required for this script:

- **NHSRdatasets** - the package containing the dataset selected (LOS_model) for this project.
- **tidyverse** - The tidyverse is a collection of R packages used in this script for most functions in this script in the manipulation of data and files.

- **here** - Used for easy file referencing in project-oriented workflows.
- **knitr** - Used in this script to visualize tables using kable().
- **caret** - Used in this script for a function called createDataPartition() which helped split the dataset into test and training sets.

Add the dictionary stuff here !!!!! !!!!!

```
library(NHSRdatasets)
library(tidyverse)
library(here)
library(knitr)
library(caret)
```

Data Dictionary

Data dictionary code here

Loading R dataset

Loading raw data into R

The following code loads the raw LOS_model data from the NHSRdatasets package into the R environment and assigns the data to a variable called LOS.

```
data(LOS_model)
LOS <- LOS_model
```

Dataset overview

Before getting the data ready for exploratory and statistical analyses, using glimpse() and head(), the dataset was briefly explored for data types, and for any obvious anomalies.

Glimpse

```
glimpse(LOS)
```

```
## Rows: 300
## Columns: 5
## $ ID           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17~
## $ Organisation <ord> Trust1, Trust2, Trust3, Trust4, Trust5, Trust6, Trust7, T~
## $ Age          <int> 55, 27, 93, 45, 70, 60, 25, 48, 51, 81, 58, 16, 21, 82, 1~
## $ LOS          <int> 2, 1, 12, 3, 11, 7, 4, 4, 7, 1, 4, 3, 1, 9, 12, 1, 4, 3, ~
## $ Death        <int> 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, ~
```

Using glimpse(), the dataset can be shown to have 300 rows and 5 columns. The R data classes of the columns can also be seen (see dictionary for more details):

- **ID** - *int* : ID values are of an integer class. This is acceptable as a row identifier and is used later for indexing.
- **Organisation** - *ord* : Organisation values is of an ordered factor class with 10 levels. This is acceptable for the data type it portrays.
- **Age** - *int* : Age values is of an integer class. This is acceptable for the data type it portrays.
- **LOS** - *int* : LOS (length of stay) values is of an integer class. This is acceptable for the data type it portrays.

- **Death** - *int* : Death values is of an integer class. Although statistical model software can work with binary 0/1 int values instead of factor, this variable is recoded into a factor with two levels for increased transparency and easier visualization (see below).

Death Recode

```
LOS <- LOS %>%
  mutate(Death=recode(factor(Death),'1'='Died','0'='Alive'))
```

The code above recodes the integer values of the raw data into a factor with two levels: “Alive” and “Died”.

Check missing values

```
LOS %>%
  map(is.na) %>%
  map(sum)
```

```
## $ID
## [1] 0
##
## $Organisation
## [1] 0
##
## $Age
## [1] 0
##
## $LOS
## [1] 0
##
## $Death
## [1] 0
```

There are no counts of missing values (NA).

Check for error values

```
summary(LOS)
```

```
##        ID          Organisation      Age             LOS           Death
##  Min.   :  1.00   Trust1 : 30   Min.   : 5.00   Min.   : 1.000   Alive:247
##  1st Qu.: 75.75   Trust2 : 30   1st Qu.:24.00   1st Qu.: 2.000   Died : 53
##  Median :150.50   Trust3 : 30   Median :54.00   Median : 4.000
##  Mean   :150.50   Trust4 : 30   Mean   :50.66   Mean   : 4.937
##  3rd Qu.:225.25   Trust5 : 30   3rd Qu.:75.25   3rd Qu.: 7.000
##  Max.   :300.00   Trust6 : 30   Max.   :95.00   Max.   :18.000
##                   (Other):120
```

The summary function shows some of the summary statistics of the dataset. Here we can confirm there are no missing values (which would be counted above). In addition, there appears to be no obvious extreme or error values (such as negative values for age or LOS).

Save raw data

As seen from the brief exploration of the dataset above, the data is extremely clean. An output of the final raw dataset is shown below of the first 10 rows of the data.

```
LOS %>%
  head(10) %>%
  kable()
```

ID	Organisation	Age	LOS	Death
1	Trust1	55	2	Alive
2	Trust2	27	1	Alive
3	Trust3	93	12	Alive
4	Trust4	45	3	Died
5	Trust5	70	11	Alive
6	Trust6	60	7	Alive
7	Trust7	25	4	Alive
8	Trust8	48	4	Alive
9	Trust9	51	7	Died
10	Trust10	81	1	Alive

With no further manipulation or transformation, the data is saved as a csv file as '/RawData/Los_raw.csv'

```
write_csv(LOS, here("RawData", "LOS_raw.csv"))
```

Splitting data into test and training sets

```
set.seed(777)
testIndex <- createDataPartition(LOS$ID, p = 10/nrow(LOS),
                                 list = FALSE,
                                 times = 1)
testIndex
```

```
##       Resample1
##  [1,]         1
##  [2,]        24
##  [3,]        60
##  [4,]        85
##  [5,]        87
##  [6,]       137
##  [7,]       159
##  [8,]       166
##  [9,]       175
## [10,]       251
## [11,]       293
## [12,]       300
```

Using the createDataPartition() function from the caret package, the ID column of the raw dataset is used as the index to split the dataset into test and training sets. The selected ID numbers are shown above to be split into the test dataset.

```
LOS_test <- LOS[testIndex,]
LOS_train <- LOS[-testIndex,]

# Save a row for assessment marking
LOS_test_marker <- LOS_test[1,]
LOS_test <- LOS_test[2:nrow(LOS_test),]
```

The LOS dataset is split using the testIndex as shown above. One observation is removed from the test data to be used for assessment marking purposes.

The final test dataset is shown in the table below.

```
LOS_test %>%
  kable()
```

ID	Organisation	Age	LOS	Death
24	Trust4	16	3	Alive
60	Trust10	12	1	Alive
85	Trust5	90	12	Alive
87	Trust7	26	3	Alive
137	Trust7	78	10	Alive
159	Trust9	53	6	Alive
166	Trust6	18	2	Alive
175	Trust5	77	3	Died
251	Trust1	73	1	Died
293	Trust3	81	8	Died
300	Trust10	93	15	Alive

The observation for marking is shown below.

```
LOS_test_marker %>%
  kable()
```

ID	Organisation	Age	LOS	Death
1	Trust1	55	2	Alive

The training table is shown below (first 10 rows)

```
LOS_train %>%
  head(10) %>%
  kable()
```

ID	Organisation	Age	LOS	Death
2	Trust2	27	1	Alive
3	Trust3	93	12	Alive
4	Trust4	45	3	Died
5	Trust5	70	11	Alive
6	Trust6	60	7	Alive
7	Trust7	25	4	Alive
8	Trust8	48	4	Alive
9	Trust9	51	7	Died
10	Trust10	81	1	Alive
11	Trust1	58	4	Alive

Data capture tool (python)

Just a description here, the main stuff will be jupyter nb / github?