Melbourne Housing

Oguzhan Demir

19/08/2023

Contents

1 Importing Dataset / Pre-analysis

We are importing our data set to studio and we save a backup version to save original data on studio. I decided to use strings as non-factor objects because i will set all the variables by myself after doing some analysis on data. Since we have some cells with "#N/A" text in it, R doesn't see them as NA values. To avoid them, we are using na.strings parameter to import them as NA's.

```r
housing_dataset <- read.csv("melbourne_data.csv",
                            stringsAsFactors = FALSE, na.strings = "#N/A")
backup <- housing_dataset
```

Before doing any cleaning and altering, we are checking structure and summary of the data to have some understanding about our data that we will analyze.

```r
str(housing_dataset)
```

```
## 'data.frame':    34857 obs. of  13 variables:
##  $ X            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Date         : chr  "3/09/2016" "3/12/2016" "4/02/2016" "4/02/2016" ...
##  $ Type         : chr  "h" "h" "h" "u" ...
##  $ Price        : chr  "NA" "1480000" "1035000" "NA" ...
##  $ Landsize     : chr  "126" "202" "156" "0" ...
##  $ BuildingArea : chr  "NA" "NA" "79" "NA" ...
##  $ Rooms        : int  2 2 2 3 3 3 4 4 2 2 ...
##  $ Bathroom     : chr  "1" "1" "1" "2" ...
##  $ Car          : chr  "1" "1" "0" "1" ...
##  $ YearBuilt    : chr  "NA" "NA" "1900" "NA" ...
##  $ Distance     : num  2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 ...
##  $ Regionname   : chr  "Northern Metropolitan" "Northern Metropolitan" "Northern Metropolitan" "Nort
##  $ Propertycount: int  4019 4019 4019 4019 4019 4019 4019 4019 4019 4019 ...
```

When we check the structure of our data set, we can see there is a column named X which is not necessary since we already have built in numbering system in R. We have lots of columns which has "chr" type which looks incorrect. Since they are character, it won't be healthy to check summary of the dataset now. So we will work with our column types now.

2 Preparing Columns for Analysis

2.1 X Column

As we mentioned above we won't use column X. According to that, we are removing that column from our dataset for further analysis.

```r
housing_dataset$X <- NULL
```

2.2 Date Column

We are formatting the "Date" column to Date object, looking into the new structure and summary to check everything is working as what we wanted. Fortunately, we had no NA values in "Date" column.

```r
housing_dataset$Date <- as.Date(housing_dataset$Date, format = "%d/%m/%Y")
```

```r
str(housing_dataset$Date)
```

```
##  Date[1:34857], format: "2016-09-03" "2016-12-03" "2016-02-04" "2016-02-04" "2017-03-04" ...
```

```r
summary(housing_dataset$Date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2016-01-28" "2016-11-19" "2017-07-08" "2017-05-23" "2017-10-28" "2018-03-17"
```

2.3 Type Column

We are renaming strings for better explanation and factorizing those values. Then, checking levels of created factor to be sure everything is alright. Looking into the new structure and summary to check everything looks like as what we wanted. We had no NA values in "Type" column too.

```
housing_dataset$Type[housing_dataset$Type == "h"] <- "House"
housing_dataset$Type[housing_dataset$Type == "u"] <- "Unit/Duplex"
housing_dataset$Type[housing_dataset$Type == "t"] <- "Townhouse"
housing_dataset$Type <- factor(housing_dataset$Type)
```

```
levels(factor(housing_dataset$Type))
```

```
## [1] "House"       "Townhouse"   "Unit/Duplex"
```

```
str(housing_dataset$Type)
```

```
##  Factor w/ 3 levels "House","Townhouse",..: 1 1 1 3 1 1 1 1 1 1 ...
```

```
summary(housing_dataset$Type)
```

```
##       House   Townhouse Unit/Duplex
##       23980        3580        7297
```

2.4 Price Column

We are formatting "Price" column to integer since we don't have any double values in Price column. Then, we are checking the structure, summary and NA values in the "Price" column. We have 7610 missing values on "Price" column.

```
housing_dataset$Price <- as.integer(housing_dataset$Price)
```

```
str(housing_dataset$Price)
```

```
##  int [1:34857] NA 1480000 1035000 NA 1465000 850000 1600000 NA NA NA ...
```

```
summary(housing_dataset$Price)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##    85000   635000   870000  1050173  1295000 11200000     7610
```

2.5 Landsize and BuildingArea Columns

We are formatting "Landsize" and "BuildingArea" columns to integer since we don't have any double values in them and they are sharing the same structures. Then, we are checking the structure, summary and NA values in those columns. We have 11810 missing values on "Landsize" and 21115 missing values on "BuildingArea" column according to summary of the column.

```r
housing_dataset$Landsize <- as.integer(housing_dataset$Landsize)
housing_dataset$BuildingArea <- as.integer(housing_dataset$BuildingArea)
```

```r
str(housing_dataset$Landsize)
```

```
##  int [1:34857] 126 202 156 0 134 94 120 400 201 202 ...
```

```r
summary(housing_dataset$Landsize)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##      0.0    224.0    521.0    593.6    670.0 433014.0    11810
```

```r
str(housing_dataset$BuildingArea)
```

```
##  int [1:34857] NA NA 79 NA 150 NA 142 220 NA NA ...
```

```r
summary(housing_dataset$BuildingArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0   102.0   136.0   160.2   188.0 44515.0   21115
```

2.6 Rooms, Bathroom, Car Columns

We will apply the same approach to "Rooms", "Bathroom" and "Car" columns since they are identical in terms of data. First of all, we will format them to integer and then we will check their structures and summaries one by one. Fortunately, we have no NA values in "Rooms" column but we have 8226 missing values in "Bathroom" and 8728 missing values in "Car" column.

```r
housing_dataset$Rooms <- as.integer(housing_dataset$Rooms)
housing_dataset$Bathroom <- as.integer(housing_dataset$Bathroom)
housing_dataset$Car <- as.integer(housing_dataset$Car)
```

```r
str(housing_dataset$Rooms)
```

```
##  int [1:34857] 2 2 2 3 3 3 4 4 2 2 ...
```

```r
summary(housing_dataset$Rooms)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   3.031   4.000  16.000
```

```r
str(housing_dataset$Bathroom)
```

```
##  int [1:34857] 1 1 1 2 2 2 1 2 1 2 ...
```

```
summary(housing_dataset$Bathroom)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   1.000   2.000   1.625   2.000  12.000    8226
```

```
str(housing_dataset$Car)
```

```
##  int [1:34857] 1 1 0 1 0 1 2 2 2 1 ...
```

```
summary(housing_dataset$Car)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   1.000   2.000   1.729   2.000  26.000    8728
```

2.7 YearBuilt Column

We will format "YearBuilt" column to integer. It will help us while dealing with outliers in the column. Since we don't have any month and day information in this column, date type won't fit to it. After formatting, we are checking the structure and summary of this column. We have 19306 missing values in this column.

```
housing_dataset$YearBuilt <- as.integer(housing_dataset$YearBuilt)
```

```
str(housing_dataset$YearBuilt)
```

```
##  int [1:34857] NA NA 1900 NA 1900 NA 2014 2006 1900 1900 ...
```

```
summary(housing_dataset$YearBuilt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1196    1940    1970    1965    2000    2106   19306
```

2.8 Distance Column

We are formatting "Distance" column to numeric since we have digits in those columns. Then we are checking structure and summary as always. We have just 1 cell missing in "Distance" column.

```
housing_dataset$Distance <- as.numeric(housing_dataset$Distance)
```

```
str(housing_dataset$Distance)
```

```
##  num [1:34857] 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 2.5 ...
```

```
summary(housing_dataset$Distance)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    6.40   10.30   11.18   14.00   48.10       1
```

2.9 Regionname Column

We are factorizing the “Regionname” column and checking its levels. Then, we look into structure and summary of it to be sure everything is correct. We have 3 cells missing in “Regionname” column.

```r
housing_dataset$Regionname <- factor(housing_dataset$Regionname)
```

```r
levels(factor(housing_dataset$Regionname))
```

```
## [1] "Eastern Metropolitan"       "Eastern Victoria"
## [3] "Northern Metropolitan"      "Northern Victoria"
## [5] "South-Eastern Metropolitan" "Southern Metropolitan"
## [7] "Western Metropolitan"       "Western Victoria"
```

```r
str(housing_dataset$Regionname)
```

```
##  Factor w/ 8 levels "Eastern Metropolitan",..: 3 3 3 3 3 3 3 3 3 3 ...
```

```r
summary(housing_dataset$Regionname)
```

```
##       Eastern Metropolitan           Eastern Victoria
##                       4377                        228
##      Northern Metropolitan          Northern Victoria
##                       9557                        203
## South-Eastern Metropolitan      Southern Metropolitan
##                       1739                      11836
##       Western Metropolitan           Western Victoria
##                       6799                        115
##                       NA's
##                          3
```

2.10 PropertyCount Column

We will factorize the “PropertyCount” column because those values actually are not different from each other. There are several property areas in different regions which has exactly the same amount of properties. So, they show us “which” property area that property belongs to. According to that, we will use that variable as factor, instead of integer. Then we are checking its structure and summary. We have 3 missing rows in “PropertyCount” column, same as “Regionname”

```r
housing_dataset$Propertycount <- factor(housing_dataset$Propertycount)
```

```r
levels(factor(housing_dataset$Propertycount))
```

Output of levels are not included since there are so many different levels. You can remove ‘results’ parameter in code block above to see output.

```r
str(housing_dataset$Propertycount)
```

```
##  Factor w/ 342 levels "83","121","129",..: 185 185 185 185 185 185 185 185 185 185 ...
```

```r
summary(housing_dataset$Propertycount)
```

```
##    21650     8870    10969    14949    10412    14577    10331    10579    11918    14887
##      844      722      583      552      491      485      467      456      444      435
##    11308    11364     8920     7809     9264    11204     6938     7485    13240     8648
##      428      424      422      420      409      405      393      378      374      371
##     8801     7717     5682     6795     6543     7082     5457     8989     6232     6482
##      369      336      319      319      304      304      293      288      285      284
##     5454     7217     7822     7570     9028    15510     5498     6567    13366     4836
##      281      278      277      262      257      255      251      249      241      237
##     5678    15321     5549     5629     4918     4675     5420     5070     6380    17496
##      237      235      234      232      228      222      212      211      210      204
##     5058     5263     3755     6763     5943     5533    13830     6244    16166    10529
##      201      199      198      197      194      186      186      184      178      176
##     3873     3284     4442     3540    10175    10926     5132     4502     3464     9758
##      175      174      173      172      172      171      167      163      162      162
##     4480     6821     3052    14092     3578     3445     4217     2947     5825     2674
##      160      159      157      157      153      151      151      150      150      149
##    14042    10999     2291     8524     2954     3265     4019     2894     2651     2671
##      146      145      144      142      140      137      137      136      135      134
##     3692    11925     4898     4380     3280    15542     2555     4553  (Other)     NA's
##      134      134      133      132      131      129      126      124     9871        3
```