Assignment 1

Steven Broussard & Genesis Roberto

The data of over 1000 eBay auctions is provided in the file eBayAcution.csv. Use RStudio to study this marketplace. (Source: The data is adapted from this book: https://www.dataminingbook.com/book/r-2nd-edition-2023 (https://www.dataminingbook.com/book/r-2nd-edition-2023))

1) Load the file: “eBayAcution.csv” and save it as auctionData.

```
auctionData = read.csv("eBayAuctions.csv")
```

2) Write a code that checks if the dataset has any missing values, a code that returns the number of auctions (i.e., rows), and one to return the number of variables (i.e., columns).

```
anyNA(auctionData)
```

```
## [1] FALSE
```

```
nrow(auctionData)
```

```
## [1] 1223
```

```
ncol(auctionData)
```

```
## [1] 6
```

3) What is the maximum auction duration? How many auctions were open for these many days? What is the average auction duration? What percentage of the auctions have an above average duration?

```
max(auctionData$Duration)
```

```
## [1] 10
```

```
sum(auctionData$Duration==10)
```

```
## [1] 220
```

```
averageduration=mean(auctionData$Duration,na.rm = T)
averageduration
```

```
## [1] 6.421096
```

```
greaterMean= (auctionData[auctionData$Duration > averageduration,])
percentage= (round((nrow(greaterMean)/nrow(auctionData)*100),digits=2))
percentage
```

```
## [1] 59.04
```

4) Create a new variable called Ratio that calculates the ratio of the closing price over the opening price for each auction and add this variable to the dataset as a new column. What’s the average ratio of all auctions? What’s the average ratio of ‘Computer’ auctions?

```
x.list= list(Ratio=(auctionData$ClosePrice/auctionData$OpenPrice))
auctionData$Ratio=x.list$Ratio
mean(x.list$Ratio,na.rm = T)
```

```
## [1] 119.6468
```

```
mean(auctionData$Ratio[auctionData$Category=="Computer"],na.rm = T)
```

```
## [1] 21.82995
```

5) Create an object named “catNames” that contains the names of unique auction categories, sorted in alphabetical order. Write a code to return the number of categories stored in this object.

```
catNames=unique(auctionData$Category)
catNames
```

```
##  [1] "Collectibles"         "Automotive"           "Computer"            
##  [4] "Electronics"          "Books"                "Music/Movie/Game"    
##  [7] "Home/Garden"          "Clothing/Accessories" "Jewelry"             
## [10] "Pottery/Glass"
```

```
sort(catNames)
```

```
##  [1] "Automotive"           "Books"                "Clothing/Accessories"
##  [4] "Collectibles"         "Computer"             "Electronics"         
##  [7] "Home/Garden"          "Jewelry"              "Music/Movie/Game"    
## [10] "Pottery/Glass"
```

```
length(catNames)
```

```
## [1] 10
```

6) Write a loop to go through “catNames” and calculate the number of auctions in each category. In so doing, save the results in a vector called “numAuctions”. Write a code to return the values stored in this object.

```
for(eachcatname in length(catNames)){
  numAuctions= unique(table(auctionData$Category))
  print(numAuctions)
}
```

```
##  [1] 167  53 118 222  33  54 101  58 398  19
```

7) Combine the two objects (catNames and numAuctions) into a new data frame called catInfo. Write two different codes to return the fifth element of the second column in the catInfo dataframe.

```
catInfo=data.frame(catNames,numAuctions)
catInfo
```

```
##                catNames numAuctions
## 1          Collectibles         167
## 2            Automotive          53
## 3              Computer         118
## 4           Electronics         222
## 5                 Books          33
## 6      Music/Movie/Game          54
## 7           Home/Garden         101
## 8  Clothing/Accessories          58
## 9               Jewelry         398
## 10        Pottery/Glass          19
```

```
catInfo[[2]][5]
```

```
## [1] 33
```

```
catInfo[["numAuctions"]][5]
```

```
## [1] 33
```

8) Write a piece of code that prints the name of each category and the number of auctions in that category.

```
for(category in 1:nrow(catInfo)){
  cat("In the category of",catInfo$catNames[category],"there was a total of",catInfo$numAuctions
[category],"auctions","\n")
}
```

```
## In the category of Collectibles there was a total of 167 auctions
## In the category of Automotive there was a total of 53 auctions
## In the category of Computer there was a total of 118 auctions
## In the category of Electronics there was a total of 222 auctions
## In the category of Books there was a total of 33 auctions
## In the category of Music/Movie/Game there was a total of 54 auctions
## In the category of Home/Garden there was a total of 101 auctions
## In the category of Clothing/Accessories there was a total of 58 auctions
## In the category of Jewelry there was a total of 398 auctions
## In the category of Pottery/Glass there was a total of 19 auctions
```

9) Create a function, called weekendTest, that checks whether a given day is a weekend (endDay of ‘Sat’ or ‘Sun’) or not and returns TRUE or FALSE (logical constants in R). Then use this function to create a new variable (called Weekend) that shows if each auction had an endDay of the weekend or not. Add this variable to the dataset as a new column. How many auction ended on weekend? (Write a code that returns this value)

```
weekendTest= function(x){
  if(x=='Sat' | x== 'Sun'){
    return(T)
  } else {
    return(F)
  }
}

weekendTest(x='Sat')
```

```
## [1] TRUE
```

```
weekendTest(x='Sun')
```

```
## [1] TRUE
```

```
weekendTest(x='Mon')
```

```
## [1] FALSE
```

```
weekend= rep(NA, nrow(auctionData))

for(i in 1: nrow(auctionData)) {
  weekend[i] = weekendTest (auctionData$endDay[i])
}
weekend
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [13]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
##  [25] FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE
##  [49]  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
##  [61]  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
##  [97] FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133]  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
## [145] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
## [157]  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
## [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [193] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## [205]  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
## [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## [229] FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE
## [241]  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
## [253]  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [265] FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
## [277] FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## [289] FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE
## [301] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [313]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
## [325]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE
## [337] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE
## [349] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## [361] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## [373]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [385]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [397]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [409]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [421]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
## [433]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE
## [445]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [457] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
## [469] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## [481] FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE
## [493]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
## [505]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [517] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE
## [529]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [541] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
## [553] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE
## [565]  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [577]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
## [589]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [601] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
## [613]  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
##  [625] FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [637] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [649] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
##  [661] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [673] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE
##  [685] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
##  [697] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [709] FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE
##  [721] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [733] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE
##  [745] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
##  [757]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
##  [769]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE
##  [781]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [793] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [805]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [817]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE
##  [829] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##  [841] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
##  [853]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [865] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [877]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [889]  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [901]  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
##  [913] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [925] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [937] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [949] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [961] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [973] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [985] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [997] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1009] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1021] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1033] FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
## [1045]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [1057]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [1069]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [1081]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
## [1093]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
## [1105] FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [1117] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE
## [1129]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
## [1141]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [1153] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## [1165]  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE
## [1177]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [1189] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## [1201]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
## [1213] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
auctionData$Weekend = weekend
sum(weekend)
```

```
## [1] 400
```