Programming 1 - Assignment 1

Your name(s): Andrew Harper and Andrew Singh

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)

---

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).

```
#Checks if the dataset has any missing values - No NAs
anyNA(auctionData)
sum(is.na(auctionData))

#Number of Auctions - 1223 Auctions
nrow(auctionData)
#Number of Variables - 6 variables
ncol(auctionData)
```

```
FALSE
0
1223
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?

```
#Maximum Auction Duration - 10
max(auctionData$Duration)

#How many auctions were open for these days - 220
nrow(auctionData[auctionData$Duration == 10,])

#Average Auction Duration - 6.421096
mean(auctionData$Duration)

#What Percentage of Auctions have an above average duration - 59.04%
(nrow(auctionData[auctionData$Duration > mean(auctionData$Duration),]) / nrow(auctionData))*100
```

```
10
220
6.42109566639411
59.0351594439902
```

---

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?

```
# Price Ratio -
auctionData$Ratio = (auctionData$ClosePrice / auctionData$OpenPrice)

#Average Price Ration for all auctions - 119.6468
mean(auctionData$Ratio)

#Average Ratio for Computer Auctions - 21.82995
computerAuctions = auctionData[auctionData$Category == "Computer",]
mean(computerAuctions$Ratio)
```

```
119.646793184339
21.8299473834876
```

---

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.

```
#Unique Categories -10
catNames <- sort(unique(auctionData$Category))
catNames
length(catNames)
```

```
'Automotive' · 'Books' · 'Clothing/Accessories' · 'Collectibles' · 'Computer' · 'Electronics' · 'Home/Garden' · 'Jewelry' · 'Music/Movie/Game' · 'Pottery/Glass'
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.

```
numAuctions <- vector()

for(i in catNames){
  subset_data <- auctionData[auctionData$Category == i,]

  # saving value in empty vector in each iteration
  numAuctions[i] <- nrow(subset_data)

}

numAuctions
```

```
Automotive: 167 Books: 53 Clothing/Accessories: 118 Collectibles: 222 Computer: 33 Electronics: 54 Home/Garden: 101
Jewelry: 58 Music/Movie/Game: 398 Pottery/Glass: 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(cbind(catNames, numAuctions))
catInfo
# Write two different codes to return the fifth element of the second column in the catInfo dataframe. - 33
catInfo[5,2]
catInfo$numAuctions[5]
```

A data.frame: 10 × 2

	catNames	numAuctions
	<chr>	<chr>
Automotive	Automotive	167
Books	Books	53
Clothing/Accessories	Clothing/Accessories	118
Collectibles	Collectibles	222
Computer	Computer	33
Electronics	Electronics	54
Home/Garden	Home/Garden	101
Jewelry	Jewelry	58
Music/Movie/Game	Music/Movie/Game	398
Pottery/Glass	Pottery/Glass	19

```
'33'
'33'
```

---

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

```
catInfo
numAuctions
```

A data.frame: 10 × 2

	catNames	numAuctions
	<chr>	<chr>
Automotive	Automotive	167
Books	Books	53
Clothing/Accessories	Clothing/Accessories	118
Collectibles	Collectibles	222
Computer	Computer	33
Electronics	Electronics	54
Home/Garden	Home/Garden	101
Jewelry	Jewelry	58
Music/Movie/Game	Music/Movie/Game	398
Pottery/Glass	Pottery/Glass	19

Automotive: 167 Books: 53 Clothing/Accessories: 118 Collectibles: 222 Computer: 33 Electronics: 54 Home/Garden: 101 Jewelry: 58 Music/Movie/Game: 398 Pottery/Glass: 19

---

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(day) {
  return(day %in% c('Sat', 'Sun'))
}

# Apply the function to create a new column 'Weekend'
auctionData$Weekend <- sapply(auctionData$endDay, weekendTest)
auctionData

# Count the number of auctions that ended on a weekend
numWeekendAuctions <- sum(auctionData$Weekend)
numWeekendAuctions
```

A data.frame: 1223 × 8

Category	sellerRating	Duration	OpenPrice	ClosePrice	endDay	Ratio	Weekend
<chr>	<int>	<int>	<dbl>	<dbl>	<chr>	<dbl>	<lgl>
Collectibles	1	5	0.010000	112.52	Tue	11252.000000	FALSE
Collectibles	1	5	0.010000	61.05	Tue	6105.000000	FALSE
Collectibles	1	5	0.010000	52.02	Tue	5202.000000	FALSE
Collectibles	1	5	0.010000	46.02	Tue	4602.000000	FALSE
Collectibles	1	5	0.010000	26.00	Tue	2600.000000	FALSE
Collectibles	1	5	0.010000	20.70	Tue	2070.000000	FALSE
Automotive	1	5	9.990000	197.50	Sat	19.769770	TRUE
Computer	1	5	1.229825	35.05	Mon	28.500000	FALSE
Collectibles	1	5	0.010000	188.33	Fri	18833.000000	FALSE
Collectibles	1	5	0.010000	43.00	Fri	4300.000000	FALSE
Collectibles	1	5	0.010000	23.65	Fri	2365.000000	FALSE
Collectibles	1	5	0.010000	22.00	Fri	2200.000000	FALSE
Electronics	4	10	1.229697	81.16	Sat	66.000000	TRUE
Electronics	4	5	0.010000	62.00	Sat	6200.000000	TRUE
Books	5	10	6.150000	6.15	Wed	1.000000	FALSE
Collectibles	5	5	19.990000	77.00	Sun	3.851926	TRUE
Electronics	7	3	67.634807	222.58	Sat	3.290909	TRUE
Music/Movie/Game	7	3	36.890000	36.89	Sat	1.000000	TRUE
Electronics	8	5	69.990000	69.99	Tue	1.000000	FALSE
Books	8	5	15.000000	47.78	Sun	3.185333	TRUE
Automotive	8	7	25.000000	91.00	Sat	3.640000	TRUE
Electronics	9	5	1.230000	1.23	Sun	1.000000	TRUE
Automotive	10	5	50.000000	170.50	Tue	3.410000	FALSE
Electronics	10	10	1.229752	197.99	Sat	161.000000	TRUE
Automotive	10	5	1.000000	70.00	Mon	70.000000	FALSE
Electronics	13	10	122.975490	250.87	Wed	2.040000	FALSE
Electronics	13	7	122.973928	278.29	Sun	2.263000	TRUE
Books	15	10	1.231076	3.09	Sun	2.510000	TRUE
Music/Movie/Game	17	7	1.766933	23.22	Thu	13.141414	FALSE
Music/Movie/Game	17	7	1.766914	18.74	Thu	10.606061	FALSE
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Music/Movie/Game	9003	7	3.99	5.51	Fri	1.380952	FALSE
Electronics	9747	5	25.00	25.00	Tue	1.000000	FALSE
Automotive	10067	3	57.00	57.00	Sat	1.000000	TRUE
Automotive	10067	3	39.50	39.50	Sat	1.000000	TRUE
Automotive	10067	3	21.00	21.00	Sat	1.000000	TRUE
Automotive	10067	3	21.00	21.00	Sat	1.000000	TRUE
Automotive	10067	3	19.50	19.50	Sat	1.000000	TRUE
Automotive	10067	3	18.00	18.50	Sat	1.027778	TRUE
Automotive	10067	3	18.00	18.00	Sat	1.000000	TRUE
Automotive	10067	3	15.00	15.00	Sat	1.000000	TRUE
Automotive	10067	3	12.99	12.99	Sat	1.000000	TRUE
Automotive	10067	3	12.50	12.50	Sat	1.000000	TRUE
Automotive	10067	3	8.50	8.50	Sat	1.000000	TRUE