Data Preprocessing

Charlie Lock

2023-03-17

Data Description

Source: https://www.kaggle.com/datasets/neuromusic/avocado-prices (https://www.kaggle.com/datasets/neuromusic/avocado-prices)

Description: A weekly scan taken in 2018 for the price and total retail volume of Hass avocados in the United States of America between the 4th of January 2015 and the 25th of March 2018. The data is taken from the cash registers of retailers around the country. Only Hass avocados are included in this data set.

Variables:

Index: Ranging from 0-51, representing a data entry taken at the start of a certain week with 0 being the week starting at the end of the calendar year and 51 being the week starting at the start of the calendar year.

Date: The date the observation was made. A week separates each data entry in this data set and range from the 4th of January 2015 to the 25th of March 2018.

AveragePrice: The average price of a single avocado over the selected time period (a week).

Total.Volume: The total number of avocados sold in the selected time period (a week).

4046: The price look-up code of a small Hass avocado. The value is the total number of these avocados sold.

4225: The price look-up code of a large Hass avocado. The value is the total number of these avocados sold.

4770: The price look-up code of a extra large Hass avocado. The value is the total number of these avocados sold.

Total Bags: Total number of bags of avocados sold (small bags and large bags included).

Small Bags: Total number of small bags of avocados sold.

Large Bags: Total number of large bags of avocados sold.

XLarge Bags: Total number of extra large bags of avocados sold.

type: The type of avocado sold. Either conventional or organic.

year: The year the observation was made

Region: The city or region of the observation

Read/Import Data

```
avocados <- read.csv("/Users/charlielock/Documents/R/Datasets/avocado.csv", stringsAs
Factors = TRUE)
head(avocados)
```

	X	Date	AveragePrice	Total.Volume	X4046	X4225	X4770	Total.Bags	Small
	<int>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
1	0	2015-12-27	1.33	64236.62	1036.74	54454.85	48.16	8696.87	86

	X	Date	AveragePrice	Total.Volume	X4046	X4225	X4770	Total.Bags	Small
	<int>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
2	1	2015-12-20	1.35	54876.98	674.28	44638.81	58.33	9505.56	94
3	2	2015-12-13	0.93	118220.22	794.70	109149.67	130.50	8145.35	80
4	3	2015-12-06	1.08	78992.15	1132.00	71976.41	72.58	5811.16	56
5	4	2015-11-29	1.28	51039.60	941.48	43838.39	75.78	6183.95	59
6	5	2015-11-22	1.26	55979.78	1184.27	48067.99	43.61	6683.91	65

6 rows | 1-10 of 15 columns

The csv file is downloaded on to the desktop from the original source and then read into R using the read.csv() function. The head() function is used to show the first 6 observations of the data set. stringsAsfactors is defaulted to FALSE when using read.csv however for for this report it needs to be set to TRUE.

Inspect and Understand

```
dim(avocados)
```

```
## [1] 18249    14
```

```
colnames(avocados)
```

```
##  [1] "X"            "Date"         "AveragePrice" "Total.Volume" "X4046"
##  [6] "X4225"        "X4770"        "Total.Bags"   "Small.Bags"   "Large.Bags"
## [11] "XLarge.Bags"  "type"         "year"         "region"
```

```
str(avocados)
```

```
## 'data.frame':    18249 obs. of  14 variables:
##  $ X           : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Date        : Factor w/ 169 levels "2015-01-04","2015-01-11",..: 52 51 50 49 48
47 46 45 44 43 ...
##  $ AveragePrice: num  1.33 1.35 0.93 1.08 1.28 1.26 0.99 0.98 1.02 1.07 ...
##  $ Total.Volume: num  64237 54877 118220 78992 51040 ...
##  $ X4046       : num  1037 674 795 1132 941 ...
##  $ X4225       : num  54455 44639 109150 71976 43838 ...
##  $ X4770       : num  48.2 58.3 130.5 72.6 75.8 ...
##  $ Total.Bags  : num  8697 9506 8145 5811 6184 ...
##  $ Small.Bags  : num  8604 9408 8042 5677 5986 ...
##  $ Large.Bags  : num  93.2 97.5 103.1 133.8 197.7 ...
##  $ XLarge.Bags : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ type        : Factor w/ 2 levels "conventional",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ year        : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
##  $ region      : Factor w/ 54 levels "Albany","Atlanta",..: 1 1 1 1 1 1 1 1 1 1
...
```

The use of the dim() function outputs the dimensions of the data frame. The colnames() function is used to output the 14 different titles of the columns/variables in the data frame. The str() function allows us to view the structures of the variables in the output.

Subsetting

```
avocado_subset <- avocados[1:10, ]
avocado_matrix <- as.matrix(avocado_subset)
avocado_matrix
```

```
##    X   Date         AveragePrice Total.Volume X4046     X4225       X4770
## 1  "0" "2015-12-27" "1.33"       " 64236.62"  "1036.74" " 54454.85" " 48.16"
## 2  "1" "2015-12-20" "1.35"       " 54876.98"  " 674.28" " 44638.81" " 58.33"
## 3  "2" "2015-12-13" "0.93"       "118220.22"  " 794.70" "109149.67" "130.50"
## 4  "3" "2015-12-06" "1.08"       " 78992.15"  "1132.00" " 71976.41" " 72.58"
## 5  "4" "2015-11-29" "1.28"       " 51039.60"  " 941.48" " 43838.39" " 75.78"
## 6  "5" "2015-11-22" "1.26"       " 55979.78"  "1184.27" " 48067.99" " 43.61"
## 7  "6" "2015-11-15" "0.99"       " 83453.76"  "1368.92" " 73672.72" " 93.26"
## 8  "7" "2015-11-08" "0.98"       "109428.33"  " 703.75" "101815.36" " 80.00"
## 9  "8" "2015-11-01" "1.02"       " 99811.42"  "1022.15" " 87315.57" " 85.34"
## 10 "9" "2015-10-25" "1.07"       " 74338.76"  " 842.40" " 64757.44" "113.00"
##    Total.Bags Small.Bags Large.Bags XLarge.Bags type           year   region
## 1  " 8696.87" " 8603.62" " 93.25"   "0"         "conventional" "2015" "Albany"
## 2  " 9505.56" " 9408.07" " 97.49"   "0"         "conventional" "2015" "Albany"
## 3  " 8145.35" " 8042.21" "103.14"   "0"         "conventional" "2015" "Albany"
## 4  " 5811.16" " 5677.40" "133.76"   "0"         "conventional" "2015" "Albany"
## 5  " 6183.95" " 5986.26" "197.69"   "0"         "conventional" "2015" "Albany"
## 6  " 6683.91" " 6556.47" "127.44"   "0"         "conventional" "2015" "Albany"
## 7  " 8318.86" " 8196.81" "122.05"   "0"         "conventional" "2015" "Albany"
## 8  " 6829.22" " 6266.85" "562.37"   "0"         "conventional" "2015" "Albany"
## 9  "11388.36" "11104.53" "283.83"   "0"         "conventional" "2015" "Albany"
## 10 " 8625.92" " 8061.47" "564.45"   "0"         "conventional" "2015" "Albany"
```

In order to subset the data frame with only the first 10 observations I define avocados[1:10,] as avocado_subset so I can convert just the first 10 observations into a matrix in the next step. avocados[1:10,] is written as is because I only want the first 10 observations but would still like to maintain all 14 variables so the second part of the function input is left blank.

Converting the newly subsetted data frame into a data matrix is the next step. The as.matrix() function is used to do this.

Create a New Data Frame

```
df_1 <- data.frame(col1 = 1:10, col2 = c ("Low", "Medium", "Low", "High", "High", "Me
dium", "Medium", "High", "Low", "Low"), stringsAsFactors = TRUE)
str(df_1)
```

```
## 'data.frame':    10 obs. of  2 variables:
##  $ col1: int  1 2 3 4 5 6 7 8 9 10
##  $ col2: Factor w/ 3 levels "High","Low","Medium": 2 3 2 1 1 3 3 1 2 2
```

```
df_1
```

col1 <int>	col2 <fct>
1	Low
2	Medium
3	Low
4	High
5	High
6	Medium
7	Medium
8	High
9	Low
10	Low

1-10 of 10 rows

```
v1 <- c(7.2, 13.5, 5.4, 23.6, 26.5, 14.6, 14.4, 28.3, 8.1, 5.8)
df2 <- cbind(df_1, v1)
df2
```

col1 <int>	col2 <fct>	v1 <dbl>
1	Low	7.2
2	Medium	13.5
3	Low	5.4
4	High	23.6
5	High	26.5
6	Medium	14.6
7	Medium	14.4
8	High	28.3
9	Low	8.1
10	Low	5.8

1-10 of 10 rows

To create data frame from scratch I use the data.frame function to create a 2 variable frame with 10 observations. Since one of the variables must be an integer variable I use the 10 integers from 1 to 10. The second variable is an ordinal variable which is a variable that is on a scale and are ordered but the explicit

difference between the scales is not outlined. I chose “Low”, “Medium” and “High” as my 3 levels. Use of the str() function confirms the structure of variables that are used in this data frame as well as the levels of the ordinal variable.

A new vector is created with numerical variables. Using the cbind() function, the new vector is then added onto the data frame that was created in the previous step (df_1) and defined as a new data frame (df2). df2 is then written down again in order to show the new data frame in the output.