2021-11-04

#1) Load the price index .csv file attached via this assignment#########################

destfile<- "D:\\One\\OneDrive\\My research\\5th semester\\R\\Assignment 4\\food-price-index-September-2021-index-numbers-csv-tables.csv"  
pricedata<- read.csv(destfile) #load the file

#2) Use 4 methods that you learned in the last two sessions to manipulate the dataset####

#2.1: read the data file and overview its content (library(data.table))  
head(pricedata,n=3) # check the first 3 rows

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Series\_reference | Period | Data\_value | STATUS | UNITS | Subject | Group | Series\_title\_1 |
| CPIM.SAP0100 | 2006.06 | 3.11 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| CPIM.SAP0100 | 2006.07 | 2.78 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| CPIM.SAP0100 | 2006.08 | 2.43 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |

tail(pricedata,n=10)# check the last 10 rows

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Series\_reference | Period | Data\_value | STATUS | UNITS | Subject | Group | Series\_title\_1 |
| 25954 | CPIM.SAP0269 | 2020.12 | 3.08 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Chewing gum, packet, each |
| 25955 | CPIM.SAP0269 | 2021.01 | 3.10 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Chewing gum, packet, each |
| 25956 | CPIM.SAP0269 | 2021.02 | 3.09 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Chewing gum, packet, each |
| 25957 | CPIM.SAP0269 | 2021.03 | 3.10 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Chewing gum, packet, each |
| 25958 | CPIM.SAP0269 | 2021.04 | 3.08 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Chewing gum, packet, each |
| 25959 | CPIM.SAP0269 | 2021.05 | 3.12 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Chewing gum, packet, each |
| 25960 | CPIM.SAP0269 | 2021.06 | 3.16 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Chewing gum, packet, each |
| 25961 | CPIM.SAP0269 | 2021.07 | 3.10 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Chewing gum, packet, each |
| 25962 | CPIM.SAP0269 | 2021.08 | 3.13 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Chewing gum, packet, each |
| 25963 | CPIM.SAP0269 | 2021.09 | 3.16 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Chewing gum, packet, each |

summary(pricedata) # summary of the object

## Series\_reference Period Data\_value STATUS   
## Length:25963 Min. :2006 Min. : 0.900 Length:25963   
## Class :character 1st Qu.:2010 1st Qu.: 2.640 Class :character   
## Mode :character Median :2014 Median : 3.660 Mode :character   
## Mean :2014 Mean : 5.432   
## 3rd Qu.:2018 3rd Qu.: 6.160   
## Max. :2021 Max. :37.960   
## NA's :82   
## UNITS Subject Group Series\_title\_1   
## Length:25963 Length:25963 Length:25963 Length:25963   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##

dim(pricedata) # check the dimension

## [1] 25963 8

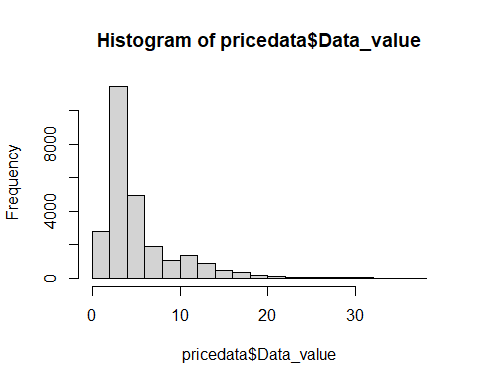
names(pricedata) # check the object names

## [1] "Series\_reference" "Period" "Data\_value" "STATUS"   
## [5] "UNITS" "Subject" "Group" "Series\_title\_1"

str(pricedata)# the structure

## 'data.frame': 25963 obs. of 8 variables:  
## $ Series\_reference: chr "CPIM.SAP0100" "CPIM.SAP0100" "CPIM.SAP0100" "CPIM.SAP0100" ...  
## $ Period : num 2006 2006 2006 2006 2006 ...  
## $ Data\_value : num 3.11 2.78 2.43 2.42 3.04 3.24 3.27 3.18 3.74 4.21 ...  
## $ STATUS : chr "FINAL" "FINAL" "FINAL" "FINAL" ...  
## $ UNITS : chr "Dollars" "Dollars" "Dollars" "Dollars" ...  
## $ Subject : chr "Consumers Price Index - CPI" "Consumers Price Index - CPI" "Consumers Price Index - CPI" "Consumers Price Index - CPI" ...  
## $ Group : chr "Food Price Index Selected Monthly Weighted Average Prices for New Zealand" "Food Price Index Selected Monthly Weighted Average Prices for New Zealand" "Food Price Index Selected Monthly Weighted Average Prices for New Zealand" "Food Price Index Selected Monthly Weighted Average Prices for New Zealand" ...  
## $ Series\_title\_1 : chr "Oranges, 1kg" "Oranges, 1kg" "Oranges, 1kg" "Oranges, 1kg" ...

#attributes(pricedata)# object's attributes  
hist(pricedata$Data\_value)# Use a histogram to display data distribution



table(pricedata$Data\_value)[1:5] # Frequency of occurrence of the first 5 values

##   
## 0.9 0.91 0.93 0.94 0.95   
## 1 1 1 1 4

is.factor(pricedata$Series\_title\_1) # Determine whether it is factor data

## [1] FALSE

#as.factor(pricedata$Series\_title\_1) # Convert to factor data  
  
#2.2: Remove the missing data  
colSums(is.na(pricedata))

## Series\_reference Period Data\_value STATUS   
## 0 0 82 0   
## UNITS Subject Group Series\_title\_1   
## 0 0 0 0

#2.2.1 Method 1: na.omit()  
good1<-na.omit(pricedata)  
dim(good1)

## [1] 25881 8

#2.2.2 Method 2: complete.cases()  
good2<-pricedata[complete.cases(pricedata),]   
dim(good2)

## [1] 25881 8

#2.2.3 Method 3: is.na()  
badrow<-which(rowSums(is.na(pricedata))>0) # Find the rows with missing values in the table "pricedata"  
bad<-pricedata[badrow,] # Save these rows with missing values in a table "bad"  
good3<-pricedata[-badrow,] # Save rows without missing values in the original table  
dim(good3)

## [1] 25881 8

#2.3: Modify table  
#2.3.1 Change the factor name  
names(good3)[1]<-"reference" # Change the factor name through the names() function  
names(good3)[1]<-"Series\_reference" #Change it back  
  
#2.3.2 Sorting  
sordata<-sort(good1$Data\_value,decreasing=TRUE)  
head(sordata)

## [1] 37.96 37.49 36.91 36.38 36.14 36.11

#2.3.3 Ordering  
#Method 1: order  
ordata<-good1[order(good1$Series\_reference,good1$Data\_value),]  
head(ordata)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Series\_reference | Period | Data\_value | STATUS | UNITS | Subject | Group | Series\_title\_1 |
| 63 | CPIM.SAP0100 | 2011.08 | 2.36 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| 99 | CPIM.SAP0100 | 2014.08 | 2.36 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| 4 | CPIM.SAP0100 | 2006.09 | 2.42 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| 3 | CPIM.SAP0100 | 2006.08 | 2.43 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| 100 | CPIM.SAP0100 | 2014.09 | 2.43 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| 16 | CPIM.SAP0100 | 2007.09 | 2.47 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |

#Method 2: library(plyr)  
library(plyr)  
head(arrange(good1,Series\_reference))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Series\_reference | Period | Data\_value | STATUS | UNITS | Subject | Group | Series\_title\_1 |
| CPIM.SAP0100 | 2006.06 | 3.11 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| CPIM.SAP0100 | 2006.07 | 2.78 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| CPIM.SAP0100 | 2006.08 | 2.43 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| CPIM.SAP0100 | 2006.09 | 2.42 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| CPIM.SAP0100 | 2006.10 | 3.04 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| CPIM.SAP0100 | 2006.11 | 3.24 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |

#2.3.4 Adding new column  
#Method 1  
newdata<-transform(good1,price=(Data\_value\*100))# Add new column named "price"  
head(newdata,n=3)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Series\_reference | Period | Data\_value | STATUS | UNITS | Subject | Group | Series\_title\_1 | price |
| CPIM.SAP0100 | 2006.06 | 3.11 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg | 311 |
| CPIM.SAP0100 | 2006.07 | 2.78 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg | 278 |
| CPIM.SAP0100 | 2006.08 | 2.43 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg | 243 |

#Method 2:   
ID<-1:25881  
df<-data.frame(ID,good3)# Add serial number column  
head(df,n=3)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Series\_reference | Period | Data\_value | STATUS | UNITS | Subject | Group | Series\_title\_1 |
| 1 | CPIM.SAP0100 | 2006.06 | 3.11 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| 2 | CPIM.SAP0100 | 2006.07 | 2.78 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| 3 | CPIM.SAP0100 | 2006.08 | 2.43 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |

#2.4: Subsetting the data set  
#2.4.1 Remove the unwanted columns  
newdata1<- good3[,-c(4:7)]# Remove columns with unique values  
head(newdata1,n=3)

|  |  |  |  |
| --- | --- | --- | --- |
| Series\_reference | Period | Data\_value | Series\_title\_1 |
| CPIM.SAP0100 | 2006.06 | 3.11 | Oranges, 1kg |
| CPIM.SAP0100 | 2006.07 | 2.78 | Oranges, 1kg |
| CPIM.SAP0100 | 2006.08 | 2.43 | Oranges, 1kg |

#2.4.2 Select the desired column with conditions  
# Method 1: Designated columns  
newdata2<-good3[,c(1:3,8)]#Specify columns 1 to 3 and column 8  
head(newdata2,n=3)

|  |  |  |  |
| --- | --- | --- | --- |
| Series\_reference | Period | Data\_value | Series\_title\_1 |
| CPIM.SAP0100 | 2006.06 | 3.11 | Oranges, 1kg |
| CPIM.SAP0100 | 2006.07 | 2.78 | Oranges, 1kg |
| CPIM.SAP0100 | 2006.08 | 2.43 | Oranges, 1kg |

# Method 2: Column containing key information  
Olives<-pricedata[pricedata$Series\_title\_1=="Olives, jar, 400g",] # All rows where Series\_reference is "Olives, jar, 400g"  
head(Olives,n=2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Series\_reference | Period | Data\_value | STATUS | UNITS | Subject | Group | Series\_title\_1 |
| 24604 | CPIM.SAP0261 | 2017.10 | 4.41 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Olives, jar, 400g |
| 24605 | CPIM.SAP0261 | 2017.11 | 4.48 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Olives, jar, 400g |

# Method 3: The column containing the specified value  
ndata<-newdata[newdata$price<=50 | newdata$price>=500,]#Columns less than or equal to 50, or greater than or equal to 500  
head(ndata,n=2)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Series\_reference | Period | Data\_value | STATUS | UNITS | Subject | Group | Series\_title\_1 | price |
| 496 | CPIM.SAP0102 | 2017.01 | 5.04 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Apples, 1kg | 504 |
| 561 | CPIM.SAP0103 | 2007.02 | 5.28 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Kiwifruit, 1kg | 528 |

# Method 4: The column containing the specified charactors  
ndata<-good3[good3$Period %in% c("2021.09"),] # %in%  
head(ndata,n=2)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Series\_reference | Period | Data\_value | STATUS | UNITS | Subject | Group | Series\_title\_1 |
| 184 | CPIM.SAP0100 | 2021.09 | 3.49 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Oranges, 1kg |
| 368 | CPIM.SAP0101 | 2021.09 | 2.92 | FINAL | Dollars | Consumers Price Index - CPI | Food Price Index Selected Monthly Weighted Average Prices for New Zealand | Bananas, 1kg |

# Method 5: Casting data frames:library(reshape2)  
library(reshape2)  
newdata3<-dcast(good3,Data\_value~Series\_reference)

## Using Series\_title\_1 as value column: use value.var to override.

## Aggregation function missing: defaulting to length

head(newdata3,n=1)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data\_value | CPIM.SAP0100 | CPIM.SAP0101 | CPIM.SAP0102 | CPIM.SAP0103 | CPIM.SAP0104 | CPIM.SAP0105 | CPIM.SAP0106 | CPIM.SAP0107 | CPIM.SAP0108 | CPIM.SAP0109 | CPIM.SAP0110 | CPIM.SAP0111 | CPIM.SAP0112 | CPIM.SAP0113 | CPIM.SAP0114 | CPIM.SAP0115 | CPIM.SAP0116 | CPIM.SAP0117 | CPIM.SAP0118 | CPIM.SAP0120 | CPIM.SAP0121 | CPIM.SAP0123 | CPIM.SAP0124 | CPIM.SAP0125 | CPIM.SAP0126 | CPIM.SAP0127 | CPIM.SAP0128 | CPIM.SAP0129 | CPIM.SAP0130 | CPIM.SAP0131 | CPIM.SAP0132 | CPIM.SAP0134 | CPIM.SAP0136 | CPIM.SAP0137 | CPIM.SAP0138 | CPIM.SAP0139 | CPIM.SAP0140 | CPIM.SAP0142 | CPIM.SAP0143 | CPIM.SAP0144 | CPIM.SAP0145 | CPIM.SAP0146 | CPIM.SAP0147 | CPIM.SAP0148 | CPIM.SAP0149 | CPIM.SAP0151 | CPIM.SAP0152 | CPIM.SAP0153 | CPIM.SAP0154 | CPIM.SAP0155 | CPIM.SAP0156 | CPIM.SAP0157 | CPIM.SAP0158 | CPIM.SAP0159 | CPIM.SAP0160 | CPIM.SAP0161 | CPIM.SAP0162 | CPIM.SAP0164 | CPIM.SAP0165 | CPIM.SAP0166 | CPIM.SAP0167 | CPIM.SAP0168 | CPIM.SAP0169 | CPIM.SAP0170 | CPIM.SAP0172 | CPIM.SAP0173 | CPIM.SAP0174 | CPIM.SAP0176 | CPIM.SAP0177 | CPIM.SAP0178 | CPIM.SAP0179 | CPIM.SAP0180 | CPIM.SAP0181 | CPIM.SAP0182 | CPIM.SAP0183 | CPIM.SAP0184 | CPIM.SAP0185 | CPIM.SAP0186 | CPIM.SAP0187 | CPIM.SAP0188 | CPIM.SAP0189 | CPIM.SAP0190 | CPIM.SAP0191 | CPIM.SAP0192 | CPIM.SAP0193 | CPIM.SAP0194 | CPIM.SAP0195 | CPIM.SAP0197 | CPIM.SAP0198 | CPIM.SAP0199 | CPIM.SAP0200 | CPIM.SAP0201 | CPIM.SAP0202 | CPIM.SAP0203 | CPIM.SAP0204 | CPIM.SAP0205 | CPIM.SAP0207 | CPIM.SAP0208 | CPIM.SAP0209 | CPIM.SAP0210 | CPIM.SAP0211 | CPIM.SAP0212 | CPIM.SAP0213 | CPIM.SAP0214 | CPIM.SAP0215 | CPIM.SAP0216 | CPIM.SAP0217 | CPIM.SAP0219 | CPIM.SAP0220 | CPIM.SAP0221 | CPIM.SAP0222 | CPIM.SAP0223 | CPIM.SAP0224 | CPIM.SAP0225 | CPIM.SAP0226 | CPIM.SAP0227 | CPIM.SAP0228 | CPIM.SAP0229 | CPIM.SAP0230 | CPIM.SAP0231 | CPIM.SAP0232 | CPIM.SAP0233 | CPIM.SAP0234 | CPIM.SAP0235 | CPIM.SAP0236 | CPIM.SAP0238 | CPIM.SAP0239 | CPIM.SAP0240 | CPIM.SAP0241 | CPIM.SAP0242 | CPIM.SAP0243 | CPIM.SAP0244 | CPIM.SAP0245 | CPIM.SAP0246 | CPIM.SAP0247 | CPIM.SAP0248 | CPIM.SAP0249 | CPIM.SAP0251 | CPIM.SAP0252 | CPIM.SAP0253 | CPIM.SAP0254 | CPIM.SAP0256 | CPIM.SAP0257 | CPIM.SAP0258 | CPIM.SAP0259 | CPIM.SAP0260 | CPIM.SAP0261 | CPIM.SAP0262 | CPIM.SAP0263 | CPIM.SAP0264 | CPIM.SAP0265 | CPIM.SAP0266 | CPIM.SAP0267 | CPIM.SAP0268 | CPIM.SAP0269 |
| 0.9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

#3)Use the factor function for column “Series\_title\_1” and get the average for each product using the price values in column “Data\_value” by sapply function####

#Method 1:  
price1<-data.frame() # create a empty dataframe  
n=1  
fac<-factor(newdata2$Series\_title\_1,ordered=TRUE) # extract the factor names  
while (n<=length(levels(fac))){  
 fac1<-newdata2[(newdata2$Series\_title\_1 %in% c(levels(fac)[n])),] # search the factor names in Series\_title\_1 one by one  
 x <- list(fac1$Data\_value) # save the search results into x  
 price1<-rbind(c(levels(fac)[n],sapply(x, FUN = mean)),price1) #using sapply to calculate the average price, then using rbind to save the product name and its price into dataframe price1  
 n=n+1  
}  
names(price1)<-c("Product","Average Price") # set the columns names  
price1 # display the final results

|  |  |
| --- | --- |
| Product | Average Price |
| Yoghurt - flavoured, 150g pottle (supermarket only), pk of 6 | 4.90027173913044 |
| Wholegrain bread, sliced, 700g | 3.47221590909091 |
| Wheatmeal bread, sliced, 700g | 2.858125 |
| Vinegar, 750ml | 2.46852272727273 |
| Two minute noodles, multipack,5 | 2.47290322580645 |
| Tuna - canned (supermarket only), 185g | 2.43489130434783 |
| Tomatoes, canned, 400g | 1.303125 |
| Tomatoes, 1kg | 6.22304347826087 |
| Tomato sauce - canned, 560g | 2.9875 |
| Tea, takeaway | 3.06651428571429 |
| Tea bags, flavoured or herbal, box of 25 | 3.11479166666667 |
| Tea bags (supermarket only), box of 100 | 4.46673913043478 |
| Takeaway muffins and buns, each | 3.32445714285714 |
| Sweets, 200g | 2.89829545454545 |
| Sultanas (supermarket only), 375g | 2.13211956521739 |
| Sugar - white, 1.5kg | 2.5729347826087 |
| Sports energy drinks, 350ml | 3.38666666666667 |
| Sports energy drinks, 250ml | 2.00816091954023 |
| Spaghetti - canned, 420g | 1.52820652173913 |
| Soy sauce, 300ml | 2.42147727272727 |
| Soy milk, unflavoured, 1 litre | 3.31993710691824 |
| Soup, canned, 500g | 3.10096590909091 |
| Soft drinks, poured | 2.70982857142857 |
| Soft drinks, 600ml | 3.55005681818182 |
| Soft drink, 1.5 litres | 2.4108152173913 |
| Sausages, 1kg | 8.86021739130435 |
| Sandwich, fresh or toasted | 4.23857923497268 |
| Salmon, imported, pink, canned, unflavoured, 210g | 3.05823863636364 |
| Salami, 100g | 3.33085227272727 |
| Salad, takeaway, vegetable, 1kg | 10.2270454545455 |
| Salad, leaf, packaged, 150g | 4.55195402298851 |
| Roasting pork, fresh, chilled or frozen, 1kg | 10.35375 |
| Roasting lamb and hogget, fresh, chilled or frozen, 1kg | 15.0443181818182 |
| Rice - long grain, white (supermarket only), 1kg | 2.41782608695652 |
| Pumpkin, 1kg | 2.62926136363636 |
| Prepared meals, frozen, 340g | 5.57732954545455 |
| Prawns, frozen, 700g | 17.3305747126437 |
| Potatoes, 1kg | 1.74945652173913 |
| Potato fries, frozen, 1kg | 3.35647727272727 |
| Potato crisps, 150g | 1.83653225806452 |
| Pork - loin chops, 1kg | 15.8085869565217 |
| Pizza, takeaway | 13.4677714285714 |
| Pizza, fresh or frozen, with any standard topping, each | 5.48782608695652 |
| Pineapple, pieces, in juice or syrup, canned, 425g | 1.77642045454545 |
| Pineapple, 1kg | 3.27767295597484 |
| Peas - frozen (supermarket only), 1kg | 2.51298913043478 |
| Pears, 1kg | 3.77738636363636 |
| Peanuts, blanched, salted, 250g | 3.37978260869565 |
| Peanut butter, not salt free, 375g | 2.85184782608696 |
| Peaches - canned (supermarket only), 410g | 1.60722826086957 |
| Pastry, frozen sheets, puff or flaky, 800g | 5.09971590909091 |
| Pasta sauces, tomato based, 500g | 2.99357954545455 |
| Parsnips, 1kg | 5.71886363636364 |
| Packaged meal, pasta and sauce, 130g | 2.55664772727273 |
| Packaged cake slice, 300g | 3.55767045454545 |
| Oranges, 1kg | 3.38483695652174 |
| Orange juice, not apple based, 1 litre - Cheapest Available | 2.66715447154472 |
| Onions, 1kg | 2.08147727272727 |
| Olives, jar, 400g | 4.35395833333333 |
| Olive oil, pure, not extra virgin or light, 1 litre | 11.87 |
| Mussels, marinated, 375g | 5.90585227272727 |
| Mussels, live, 1kg | 3.90267045454545 |
| Mushrooms, 1kg | 11.0333152173913 |
| Muesli/cereal bars, 200g | 2.94754716981132 |
| Muesli, natural or toasted, 750g | 5.28607954545455 |
| Mixed vegetables, frozen, 1kg | 3.40534090909091 |
| Milk, calcium enriched, 2 litres | 5.10647727272727 |
| Milk - standard homogenised, 2 litres | 3.38614130434783 |
| Meat pies, chilled, 6 or 8 pack - Cheapest Available | 6.06869318181818 |
| Meat pie - hot, each | 3.70109289617486 |
| Mayonnaise, 380ml | 3.32630434782609 |
| Margarine/table spread, 500g | 2.33983695652174 |
| Mandarins, 1kg | 5.26215909090909 |
| Lettuce, 1kg | 4.39777173913043 |
| Lamb - chops, 1kg | 14.21875 |
| Kumara, 1kg | 5.29005681818182 |
| Kiwifruit, 1kg | 3.73782608695652 |
| Jam, 375g | 2.61482954545455 |
| Infant formula, 900g | 19.2929545454545 |
| Ice cream novelty, chocolate coated, each | 3.18278409090909 |
| Ice cream bought in bulk, 2 litres | 5.58818181818182 |
| Ice block, water based, each | 2.12897727272727 |
| Hummus dip, 200g | 3.74578616352201 |
| Hot chips, hot wedges | 2.97497142857143 |
| Honey, clover, creamed, 500g | 7.0625 |
| Ham, sliced or shaved, 1kg | 13.6776086956522 |
| Grapes, green or red | 7.47982954545455 |
| Fruit juice - apple based (supermarket only), 3 litre | 4.17677419354839 |
| Fruit flavoured drink powder, multipack of 3 to 5 | 1.2555625 |
| Fried and other takeaway chicken, 5 pieces | 11.3387428571429 |
| Fresh pasta, tortellini or other filled type, 300g | 4.59138364779874 |
| Fresh herbs, packaged, chilled | 3.79395833333333 |
| Fresh fish, 1kg | 29.3965340909091 |
| Flour - white (supermarket only), 1.5kg | 1.9429347826087 |
| Flat bread - pita, tortilla, or other type | 4.13422764227642 |
| Fish fillets, frozen, multipack, 500g | 7.32454545454545 |
| Fish and chips, One fish/chips | 5.94857923497268 |
| Eggs, free range, 6 pack | 4.71025157232704 |
| Eggs, dozen | 3.73086956521739 |
| Drinking chocolate, 300g | 3.92744318181818 |
| Dried pasta, spaghetti or other type, 500g | 1.88415094339623 |
| Dried mixed herbs, 10g to 15g | 2.36358695652174 |
| Dessert, frozen, 500g | 6.35062893081761 |
| Cucumber, 1kg | 7.79380681818182 |
| Cream, 300ml - Cheapest Available | 2.23369318181818 |
| Courgettes, 1kg | 8.75363636363636 |
| Corned beef, fresh, chilled or frozen, 1kg | 9.55392045454546 |
| Corn flakes, 500g | 3.38647727272727 |
| Cookie, takeaway, each | 1.87668571428571 |
| Coffee, takeway, each | 3.57508571428571 |
| Coffee, ground, 200g | 6.31471590909091 |
| Coffee - instant, 100g | 5.55038043478261 |
| Chocolate, boxed, loose, 250g | 8.46767045454545 |
| Chocolate novelty bars, 50g | 1.42545454545455 |
| Chocolate blocks, convenience stores, 100g to 250g | 4.53426136363636 |
| Chocolate - block (supermarket only), 250g | 3.9604347826087 |
| Chilled fruit juice or smoothies, 1 to 1.5 litre | 4.58635220125786 |
| Chicken, whole, frozen, No. 15 - Cheapest Available | 8.09931818181818 |
| Chicken, cooked, whole, No. 15 - Cheapest Available | 11.5759748427673 |
| Chicken pieces (excluding breast), boneless or bone in, 1kg | 8.14747126436782 |
| Chicken nuggets, frozen, 1kg | 11.0632520325203 |
| Chicken breast, 1kg | 13.9584090909091 |
| Chewing gum, packet, each | 2.7241875 |
| Cheese, processed slices, 250g | 3.50022727272727 |
| Cheese, camembert, 125g | 4.27681818181818 |
| Cheese - mild cheddar (supermarket only), 1kg | 9.10239130434783 |
| Celery, 1kg | 3.31965909090909 |
| Cauliflower, 1kg | 3.42727272727273 |
| Carrots, 1kg | 2.13630434782609 |
| Capsicums, green, else red, 1kg | 12.7696022727273 |
| Cakes and biscuits, takeaway | 3.63411428571429 |
| Cabbage, 1kg | 1.97684782608696 |
| Butter - salted, 500g | 3.97826086956522 |
| Burger, with or without accompaniments, each | 4.5976 |
| Broccoli, 1kg | 5.91711956521739 |
| Breakfast drink, 250ml, 6 pack | 7.71287356321839 |
| Breakfast biscuits, 1kg | 5.62391304347826 |
| Bread rolls, hamburger buns, 6 pack | 2.78448863636364 |
| Bread rolls, filled, hot, each | 6.27622857142857 |
| Bread - white sliced loaf, 600g | 1.32326086956522 |
| Bottled water, 750ml | 2.02994565217391 |
| Biscuits, savoury, crackers 250g | 3.160625 |
| Biscuits, plain (eg arrowroot, ginger, malt, wine), 250g | 2.21698863636364 |
| Biscuits - chocolate, 200g | 2.83195652173913 |
| Berries, frozen, 500g | 6.49983739837398 |
| Beef steak - porterhouse/sirloin, 1kg | 26.2638586956522 |
| Beef steak - blade, 1kg | 15.6879347826087 |
| Beef - mince, 1kg | 12.6989130434783 |
| Beans, 1kg | 12.8588636363636 |
| Bananas, 1kg | 2.74076086956522 |
| Bacon - middle rashers (supermarket only), 700g | 12.0044318181818 |
| Baby food, 110g | 1.09664772727273 |
| Avocado, 1kg | 9.78926136363636 |
| Apricots, dried, 100g | 2.19308943089431 |
| Apples, 1kg | 2.83760869565217 |

#Method 2:  
splitmean <- function(newdata2) { #build a function by split and sapply function  
 s <- split( newdata2, newdata2$Series\_title\_1) # split the data by Series\_title\_1  
 sapply( s, function(x) mean(x$Data\_value) )# calculate the average price  
}  
price<-splitmean(newdata2) # call the function  
price # display the final results

## Apples, 1kg   
## 2.837609   
## Apricots, dried, 100g   
## 2.193089   
## Avocado, 1kg   
## 9.789261   
## Baby food, 110g   
## 1.096648   
## Bacon - middle rashers (supermarket only), 700g   
## 12.004432   
## Bananas, 1kg   
## 2.740761   
## Beans, 1kg   
## 12.858864   
## Beef - mince, 1kg   
## 12.698913   
## Beef steak - blade, 1kg   
## 15.687935   
## Beef steak - porterhouse/sirloin, 1kg   
## 26.263859   
## Berries, frozen, 500g   
## 6.499837   
## Biscuits - chocolate, 200g   
## 2.831957   
## Biscuits, plain (eg arrowroot, ginger, malt, wine), 250g   
## 2.216989   
## Biscuits, savoury, crackers 250g   
## 3.160625   
## Bottled water, 750ml   
## 2.029946   
## Bread - white sliced loaf, 600g   
## 1.323261   
## Bread rolls, filled, hot, each   
## 6.276229   
## Bread rolls, hamburger buns, 6 pack   
## 2.784489   
## Breakfast biscuits, 1kg   
## 5.623913   
## Breakfast drink, 250ml, 6 pack   
## 7.712874   
## Broccoli, 1kg   
## 5.917120   
## Burger, with or without accompaniments, each   
## 4.597600   
## Butter - salted, 500g   
## 3.978261   
## Cabbage, 1kg   
## 1.976848   
## Cakes and biscuits, takeaway   
## 3.634114   
## Capsicums, green, else red, 1kg   
## 12.769602   
## Carrots, 1kg   
## 2.136304   
## Cauliflower, 1kg   
## 3.427273   
## Celery, 1kg   
## 3.319659   
## Cheese - mild cheddar (supermarket only), 1kg   
## 9.102391   
## Cheese, camembert, 125g   
## 4.276818   
## Cheese, processed slices, 250g   
## 3.500227   
## Chewing gum, packet, each   
## 2.724188   
## Chicken breast, 1kg   
## 13.958409   
## Chicken nuggets, frozen, 1kg   
## 11.063252   
## Chicken pieces (excluding breast), boneless or bone in, 1kg   
## 8.147471   
## Chicken, cooked, whole, No. 15 - Cheapest Available   
## 11.575975   
## Chicken, whole, frozen, No. 15 - Cheapest Available   
## 8.099318   
## Chilled fruit juice or smoothies, 1 to 1.5 litre   
## 4.586352   
## Chocolate - block (supermarket only), 250g   
## 3.960435   
## Chocolate blocks, convenience stores, 100g to 250g   
## 4.534261   
## Chocolate novelty bars, 50g   
## 1.425455   
## Chocolate, boxed, loose, 250g   
## 8.467670   
## Coffee - instant, 100g   
## 5.550380   
## Coffee, ground, 200g   
## 6.314716   
## Coffee, takeway, each   
## 3.575086   
## Cookie, takeaway, each   
## 1.876686   
## Corn flakes, 500g   
## 3.386477   
## Corned beef, fresh, chilled or frozen, 1kg   
## 9.553920   
## Courgettes, 1kg   
## 8.753636   
## Cream, 300ml - Cheapest Available   
## 2.233693   
## Cucumber, 1kg   
## 7.793807   
## Dessert, frozen, 500g   
## 6.350629   
## Dried mixed herbs, 10g to 15g   
## 2.363587   
## Dried pasta, spaghetti or other type, 500g   
## 1.884151   
## Drinking chocolate, 300g   
## 3.927443   
## Eggs, dozen   
## 3.730870   
## Eggs, free range, 6 pack   
## 4.710252   
## Fish and chips, One fish/chips   
## 5.948579   
## Fish fillets, frozen, multipack, 500g   
## 7.324545   
## Flat bread - pita, tortilla, or other type   
## 4.134228   
## Flour - white (supermarket only), 1.5kg   
## 1.942935   
## Fresh fish, 1kg   
## 29.396534   
## Fresh herbs, packaged, chilled   
## 3.793958   
## Fresh pasta, tortellini or other filled type, 300g   
## 4.591384   
## Fried and other takeaway chicken, 5 pieces   
## 11.338743   
## Fruit flavoured drink powder, multipack of 3 to 5   
## 1.255562   
## Fruit juice - apple based (supermarket only), 3 litre   
## 4.176774   
## Grapes, green or red   
## 7.479830   
## Ham, sliced or shaved, 1kg   
## 13.677609   
## Honey, clover, creamed, 500g   
## 7.062500   
## Hot chips, hot wedges   
## 2.974971   
## Hummus dip, 200g   
## 3.745786   
## Ice block, water based, each   
## 2.128977   
## Ice cream bought in bulk, 2 litres   
## 5.588182   
## Ice cream novelty, chocolate coated, each   
## 3.182784   
## Infant formula, 900g   
## 19.292955   
## Jam, 375g   
## 2.614830   
## Kiwifruit, 1kg   
## 3.737826   
## Kumara, 1kg   
## 5.290057   
## Lamb - chops, 1kg   
## 14.218750   
## Lettuce, 1kg   
## 4.397772   
## Mandarins, 1kg   
## 5.262159   
## Margarine/table spread, 500g   
## 2.339837   
## Mayonnaise, 380ml   
## 3.326304   
## Meat pie - hot, each   
## 3.701093   
## Meat pies, chilled, 6 or 8 pack - Cheapest Available   
## 6.068693   
## Milk - standard homogenised, 2 litres   
## 3.386141   
## Milk, calcium enriched, 2 litres   
## 5.106477   
## Mixed vegetables, frozen, 1kg   
## 3.405341   
## Muesli, natural or toasted, 750g   
## 5.286080   
## Muesli/cereal bars, 200g   
## 2.947547   
## Mushrooms, 1kg   
## 11.033315   
## Mussels, live, 1kg   
## 3.902670   
## Mussels, marinated, 375g   
## 5.905852   
## Olive oil, pure, not extra virgin or light, 1 litre   
## 11.870000   
## Olives, jar, 400g   
## 4.353958   
## Onions, 1kg   
## 2.081477   
## Orange juice, not apple based, 1 litre - Cheapest Available   
## 2.667154   
## Oranges, 1kg   
## 3.384837   
## Packaged cake slice, 300g   
## 3.557670   
## Packaged meal, pasta and sauce, 130g   
## 2.556648   
## Parsnips, 1kg   
## 5.718864   
## Pasta sauces, tomato based, 500g   
## 2.993580   
## Pastry, frozen sheets, puff or flaky, 800g   
## 5.099716   
## Peaches - canned (supermarket only), 410g   
## 1.607228   
## Peanut butter, not salt free, 375g   
## 2.851848   
## Peanuts, blanched, salted, 250g   
## 3.379783   
## Pears, 1kg   
## 3.777386   
## Peas - frozen (supermarket only), 1kg   
## 2.512989   
## Pineapple, 1kg   
## 3.277673   
## Pineapple, pieces, in juice or syrup, canned, 425g   
## 1.776420   
## Pizza, fresh or frozen, with any standard topping, each   
## 5.487826   
## Pizza, takeaway   
## 13.467771   
## Pork - loin chops, 1kg   
## 15.808587   
## Potato crisps, 150g   
## 1.836532   
## Potato fries, frozen, 1kg   
## 3.356477   
## Potatoes, 1kg   
## 1.749457   
## Prawns, frozen, 700g   
## 17.330575   
## Prepared meals, frozen, 340g   
## 5.577330   
## Pumpkin, 1kg   
## 2.629261   
## Rice - long grain, white (supermarket only), 1kg   
## 2.417826   
## Roasting lamb and hogget, fresh, chilled or frozen, 1kg   
## 15.044318   
## Roasting pork, fresh, chilled or frozen, 1kg   
## 10.353750   
## Salad, leaf, packaged, 150g   
## 4.551954   
## Salad, takeaway, vegetable, 1kg   
## 10.227045   
## Salami, 100g   
## 3.330852   
## Salmon, imported, pink, canned, unflavoured, 210g   
## 3.058239   
## Sandwich, fresh or toasted   
## 4.238579   
## Sausages, 1kg   
## 8.860217   
## Soft drink, 1.5 litres   
## 2.410815   
## Soft drinks, 600ml   
## 3.550057   
## Soft drinks, poured   
## 2.709829   
## Soup, canned, 500g   
## 3.100966   
## Soy milk, unflavoured, 1 litre   
## 3.319937   
## Soy sauce, 300ml   
## 2.421477   
## Spaghetti - canned, 420g   
## 1.528207   
## Sports energy drinks, 250ml   
## 2.008161   
## Sports energy drinks, 350ml   
## 3.386667   
## Sugar - white, 1.5kg   
## 2.572935   
## Sultanas (supermarket only), 375g   
## 2.132120   
## Sweets, 200g   
## 2.898295   
## Takeaway muffins and buns, each   
## 3.324457   
## Tea bags (supermarket only), box of 100   
## 4.466739   
## Tea bags, flavoured or herbal, box of 25   
## 3.114792   
## Tea, takeaway   
## 3.066514   
## Tomato sauce - canned, 560g   
## 2.987500   
## Tomatoes, 1kg   
## 6.223043   
## Tomatoes, canned, 400g   
## 1.303125   
## Tuna - canned (supermarket only), 185g   
## 2.434891   
## Two minute noodles, multipack,5   
## 2.472903   
## Vinegar, 750ml   
## 2.468523   
## Wheatmeal bread, sliced, 700g   
## 2.858125   
## Wholegrain bread, sliced, 700g   
## 3.472216   
## Yoghurt - flavoured, 150g pottle (supermarket only), pk of 6   
## 4.900272

#4) Push the r file into your GitHub like before and submit your GitHub link like prior assignments####

When you read this, I have finished uploading.

Thanks for your patience!

THE END