INNER_join_exam.R

r2025562

2023-05-03

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
#QUESTION ONE
MKmart <- read_csv("sammyR/MKmart_raw.csv")
```

```
## Rows: 6435 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Date
## dbl (7): Store, Weekly_Sales, Holiday_Flag, Temperature, Fuel_Price, CPI, Un...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#function one : structure of the dataset
str(MKmart)#date is character and all others are numerical
```

```
## spc_tbl_ [6,435 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Store       : num [1:6435] 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date        : chr [1:6435] "05/2/2010" "12/2/2010" "19/2/2010" "26/2/2010" ...
##  $ Weekly_Sales: num [1:6435] 1643691 1641957 1611968 1409728 1554807 ...
##  $ Holiday_Flag: num [1:6435] 0 1 0 0 0 0 0 0 0 0 ...
##  $ Temperature : num [1:6435] 42.3 38.5 39.9 46.6 46.5 ...
##  $ Fuel_Price  : num [1:6435] 2.57 2.55 2.51 2.56 2.62 ...
##  $ CPI         : num [1:6435] 211 211 211 211 211 ...
##  $ Unemployment: num [1:6435] 8.11 8.11 8.11 8.11 8.11 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Store = col_double(),
##   ..   Date = col_character(),
##   ..   Weekly_Sales = col_double(),
##   ..   Holiday_Flag = col_double(),
##   ..   Temperature = col_double(),
```

```
##   ..   Fuel_Price = col_double(),
##   ..   CPI = col_double(),
##   ..   Unemployment = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
dim(MKmart)#6435 rows and 8 columns
```

```
## [1] 6435    8
```

```r
summary(MKmart)#min mean median 1st and 3rd quartiles
```

```
##      Store        Date            Weekly_Sales     Holiday_Flag
##  Min.   : 1   Length:6435        Min.   : 209986   Min.   :0.00000
##  1st Qu.:12   Class :character   1st Qu.: 551601   1st Qu.:0.00000
##  Median :23   Mode  :character   Median : 957072   Median :0.00000
##  Mean   :23                      Mean   :1043994   Mean   :0.06993
##  3rd Qu.:34                      3rd Qu.:1415679   3rd Qu.:0.00000
##  Max.   :45                      Max.   :3818686   Max.   :1.00000
##                                  NA's   :37
##   Temperature       Fuel_Price         CPI         Unemployment
##  Min.   : -2.06   Min.   :2.472   Min.   :126.1   Min.   : 3.879
##  1st Qu.: 47.42   1st Qu.:2.936   1st Qu.:131.6   1st Qu.: 6.891
##  Median : 62.63   Median :3.452   Median :182.4   Median : 7.874
##  Mean   : 60.65   Mean   :3.361   Mean   :171.2   Mean   : 7.999
##  3rd Qu.: 74.94   3rd Qu.:3.735   3rd Qu.:212.2   3rd Qu.: 8.622
##  Max.   :100.14   Max.   :4.468   Max.   :227.2   Max.   :14.313
##  NA's   :7        NA's   :26      NA's   :52
```

```r
#2. Determine the variables with missing values (NAs) and print the total number
#of NAs in each of the variable.
#is.na(MKmart)
sum(is.na(MKmart))#122 missing values
```

```
## [1] 122
```

```r
colnames(MKmart)
```

```
## [1] "Store"        "Date"         "Weekly_Sales" "Holiday_Flag" "Temperature"
## [6] "Fuel_Price"   "CPI"          "Unemployment"
```

```r
sum(is.na(MKmart$Store))#zero null
```

```
## [1] 0
```

```r
sum(is.na(MKmart$CPI))#52 null values
```

```
## [1] 52
```

```r
sum(is.na(MKmart$Date))#zero null
```

```
## [1] 0
```

```r
sum(is.na(MKmart$Unemployment))#zero null
```

```
## [1] 0
```

```r
sum(is.na(MKmart$Weekly_Sales))#37null
```

```
## [1] 37
```

```r
sum(is.na(MKmart$Holiday_Flag))#zero null
```

```
## [1] 0
```

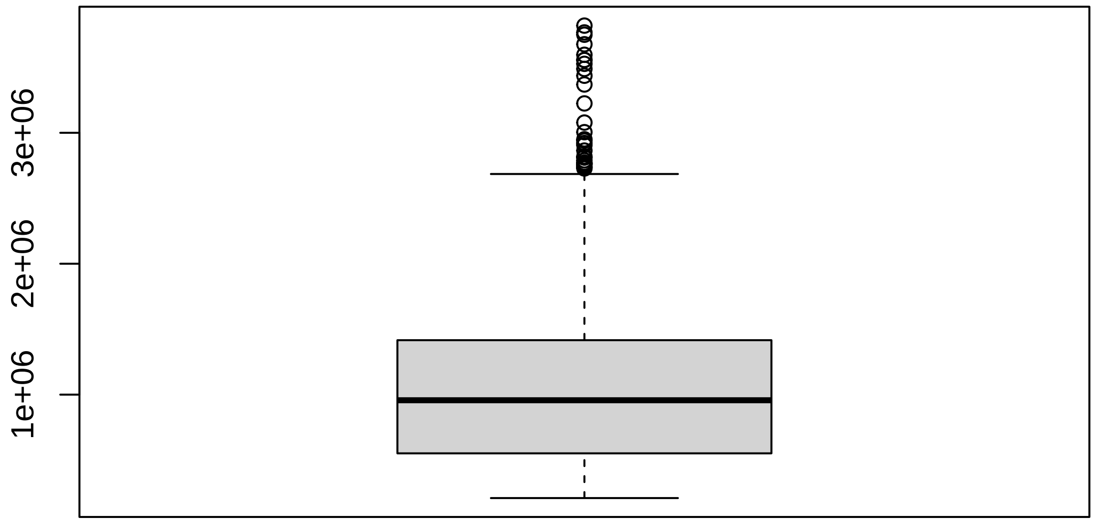
```r
sum(is.na(MKmart$Temperature))#7 null
```

```
## [1] 7
```

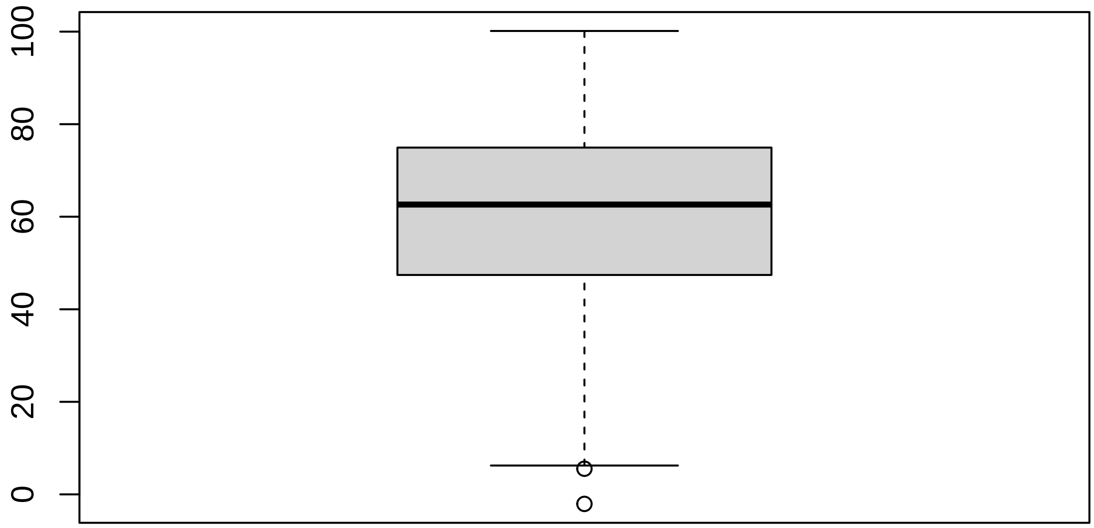
```r
sum(is.na(MKmart$Fuel_Price))#26 null
```

```
## [1] 26
```

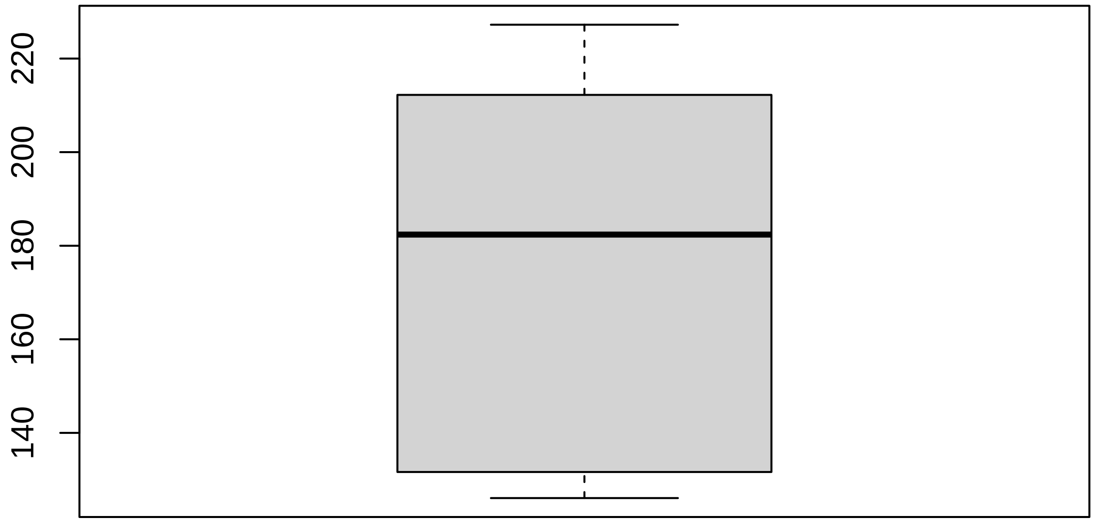
```r
#3. Determine the outliers in Weekly_Sales, Temperature, Fuel_Price, CPI and
#Unemployment variables and remove all the outliers. Make sure at the end,
#you must produce a dataframe named "MKmart2" without outliers in all those
#four variables.

#checking for outliers using a boxplot
boxplot(MKmart$Weekly_Sales)#some outlier present
```



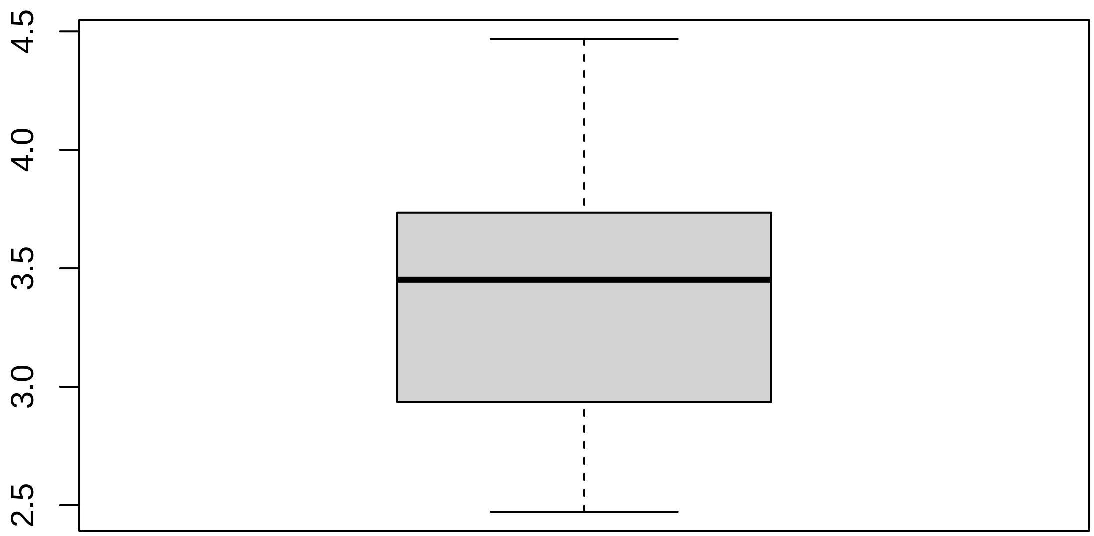

```r
boxplot(MKmart$Temperature)#some outlier present
```



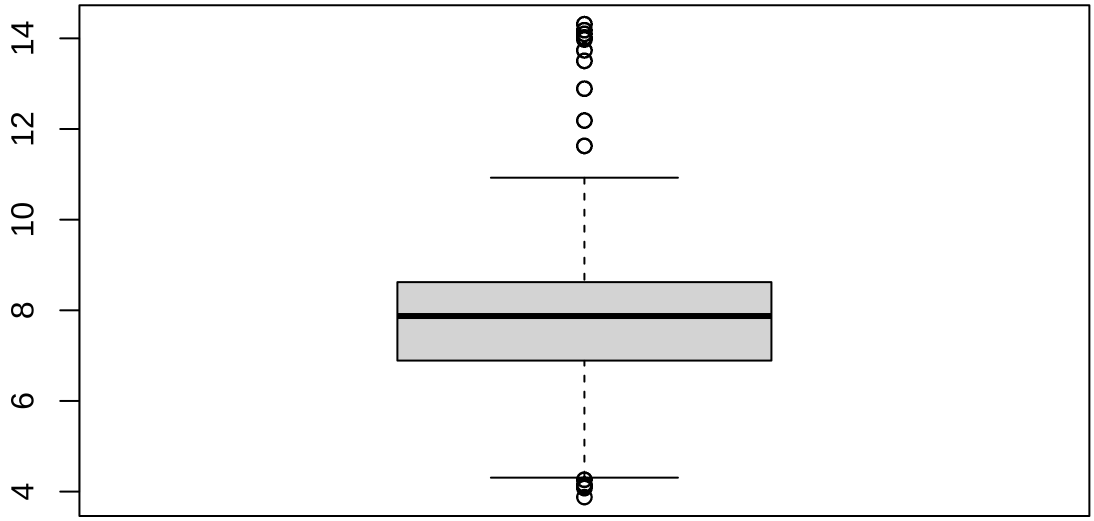
```
boxplot(MKmart$CPI)#no outlier
```




```
boxplot(MKmart$Fuel_Price)#no outlier
```




```
boxplot(MKmart$Unemployment)#some outlier present
```




```
#removing outliers in weekly sales
summary(MKmart$Weekly_Sales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  209986  551601  957072 1043994 1415679 3818686      37
```

```
IQR_sales = 1415679 - 551601
up_sale = 1415679 + 1.5*IQR_sales
low_sale = 551601 - 1.5*IQR_sales
up_sale#2711796
```

```
## [1] 2711796
```

```
#removing outliers in Temperature
summary(MKmart$Temperature)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   -2.06   47.42   62.63   60.65   74.94  100.14       7
```

```
IQR_temp = 74.94 - 47.42
up_temp = 74.94 + 1.5*IQR_temp
low_temp = 47.42  - 1.5*IQR_temp

up_temp#116.22
```

```
## [1] 116.22
```

```
#removing outliers in
summary(MKmart$Unemployment)
```

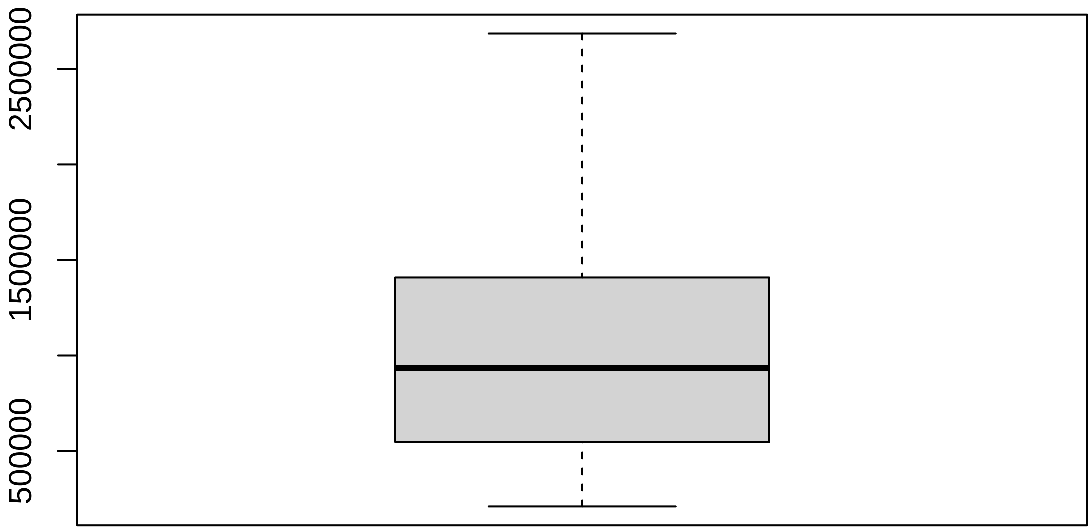
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   3.879   6.891   7.874   7.999   8.622  14.313
```

```
IQR_un = 8.622 - 6.891
up_une = 8.622 + 1.5*IQR_un
low_une = 6.891  - 1.5*IQR_un


summary(MKmart)
```

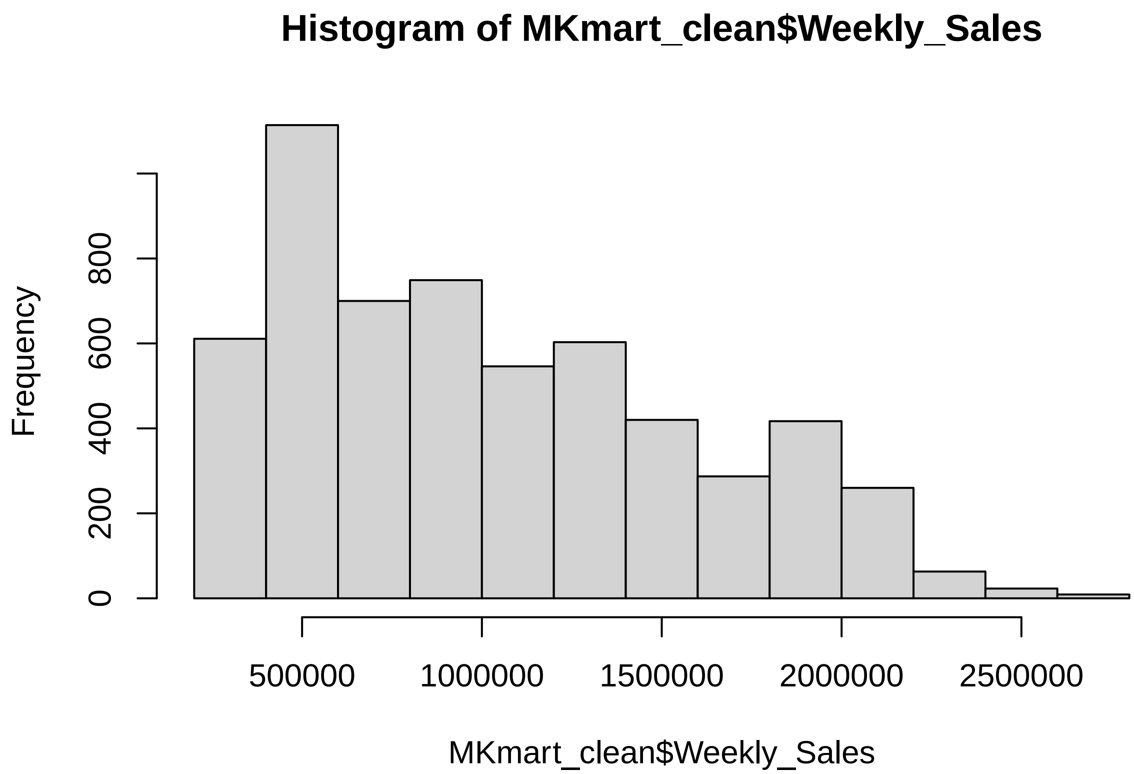
```
##      Store         Date              Weekly_Sales       Holiday_Flag
```

```
##   Min.   : 1   Length:6435        Min.   : 209986   Min.   :0.00000
##   1st Qu.:12   Class :character   1st Qu.: 551601   1st Qu.:0.00000
##   Median :23   Mode  :character   Median : 957072   Median :0.00000
##   Mean   :23                      Mean   :1043994   Mean   :0.06993
##   3rd Qu.:34                      3rd Qu.:1415679   3rd Qu.:0.00000
##   Max.   :45                      Max.   :3818686   Max.   :1.00000
##                                   NA's   :37
##    Temperature       Fuel_Price         CPI          Unemployment
##   Min.   : -2.06   Min.   :2.472   Min.   :126.1   Min.   : 3.879
##   1st Qu.: 47.42   1st Qu.:2.936   1st Qu.:131.6   1st Qu.: 6.891
##   Median : 62.63   Median :3.452   Median :182.4   Median : 7.874
##   Mean   : 60.65   Mean   :3.361   Mean   :171.2   Mean   : 7.999
##   3rd Qu.: 74.94   3rd Qu.:3.735   3rd Qu.:212.2   3rd Qu.: 8.622
##   Max.   :100.14   Max.   :4.468   Max.   :227.2   Max.   :14.313
##   NA's   :7        NA's   :26      NA's   :52
```

```
MKmart2 = subset(MKmart,Unemployment<=11.2185 & Unemployment>=low_une
                  & Temperature<=116.22 & Temperature>=low_temp &
                     Weekly_Sales<=2711796 & Weekly_Sales>=low_sale & CPI<=227.2
                   & Fuel_Price<=4.468)
boxplot(MKmart2$Weekly_Sales)#no outliers
```

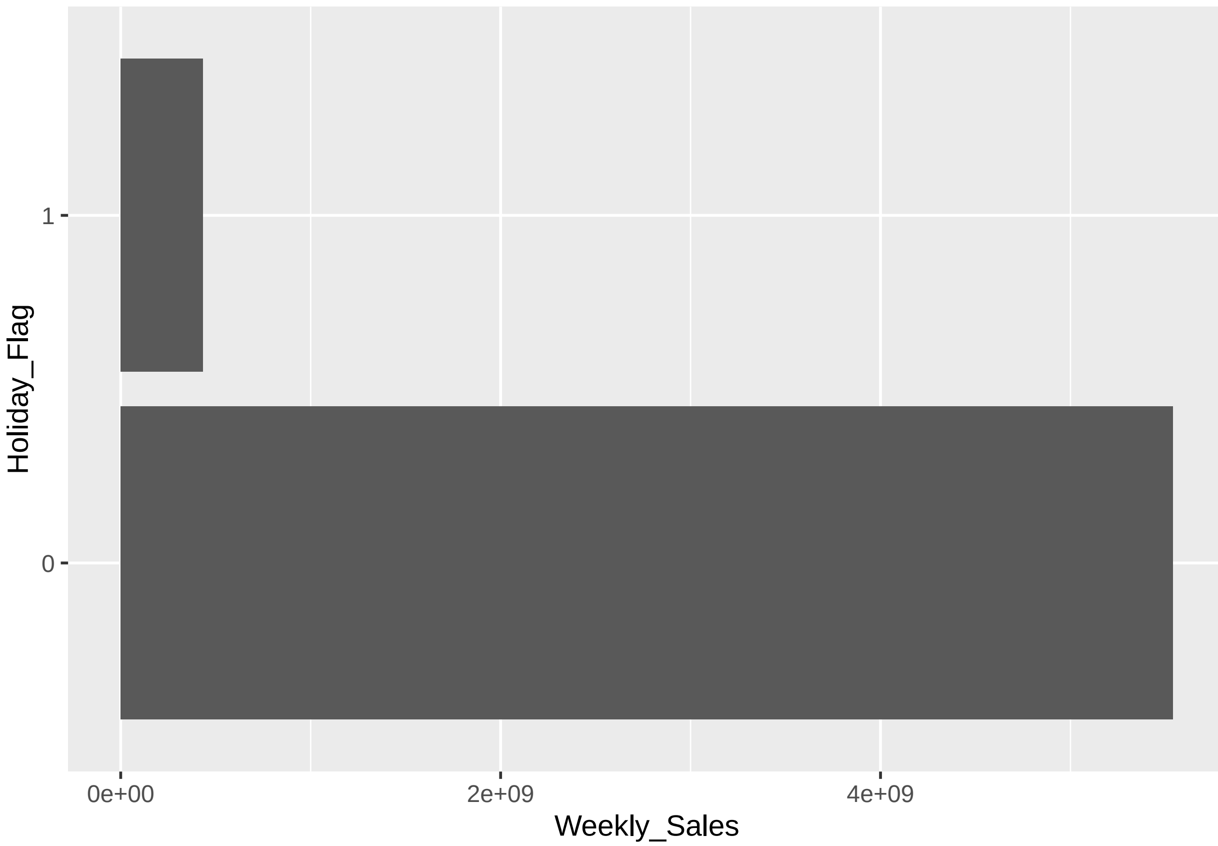



```
#4. Remove all the rows with NAs in "MKmart2" and assign a new name to the
#dataframe as "MKmart_clean".
MKmart_clean <- na.omit(MKmart2)
#is.na(MKmart_clean)

#5. Visualize the distribution of the continuous variable Weekly_Sales in the
#"MKmart_clean" dataframe, using a histogram function from ggplot2 R
#package. Add title and x-axis label to the histogram.
library(ggplot2)
hist(MKmart_clean$Weekly_Sales)
```




```r
#6. Create a bar chart using the ggplot2 R package to visualize the comparison
#between Holiday_Flag and Weekly_Sales, based on the data in the
#"MKmart_clean" d
MKmart_clean$Holiday_Flag <- as.factor(MKmart_clean$Holiday_Flag)
ggplot(MKmart_clean, aes(x=Weekly_Sales, y=Holiday_Flag)) +
  geom_bar(stat = "identity")+
  scale_fill_brewer(palette = "steelblue") +
  theme(legend.position="none")
```

```
## Warning in pal_name(palette, type): Unknown palette steelblue
```




```
#7. Interpret the relationship between Holiday_Flag and Weekly_Sales in your
#own words.
  #days without holiday had more sales compared to those with



#8. Using the ggplot2 R package, create a correlation heatmap with correlation
#oefficient labels (2 decimal places) to evaluate the relationship between
#Weekly Sales, Temperature, and Fuel_Price
tmp <- MKmart_clean %>%
  dplyr::select('Weekly_Sales','Temperature','Fuel_Price')
head(tmp)
```

```
## # A tibble: 6 x 3
##   Weekly_Sales Temperature Fuel_Price
##          <dbl>       <dbl>      <dbl>
## 1     1643691.        42.3       2.57
## 2     1641957.        38.5       2.55
## 3     1611968.        39.9       2.51
## 4     1409728.        46.6       2.56
## 5     1554807.        46.5       2.62
## 6     1439542.        57.8       2.67
```

```
install.packages("lattice")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
library(lattice)

# rounding to 2 decimal places
corr_m <- round(cor(tmp),2)
head(corr_m)
```

```
##              Weekly_Sales Temperature Fuel_Price
## Weekly_Sales         1.00       -0.05       0.02
## Temperature         -0.05        1.00       0.15
## Fuel_Price           0.02        0.15       1.00
```

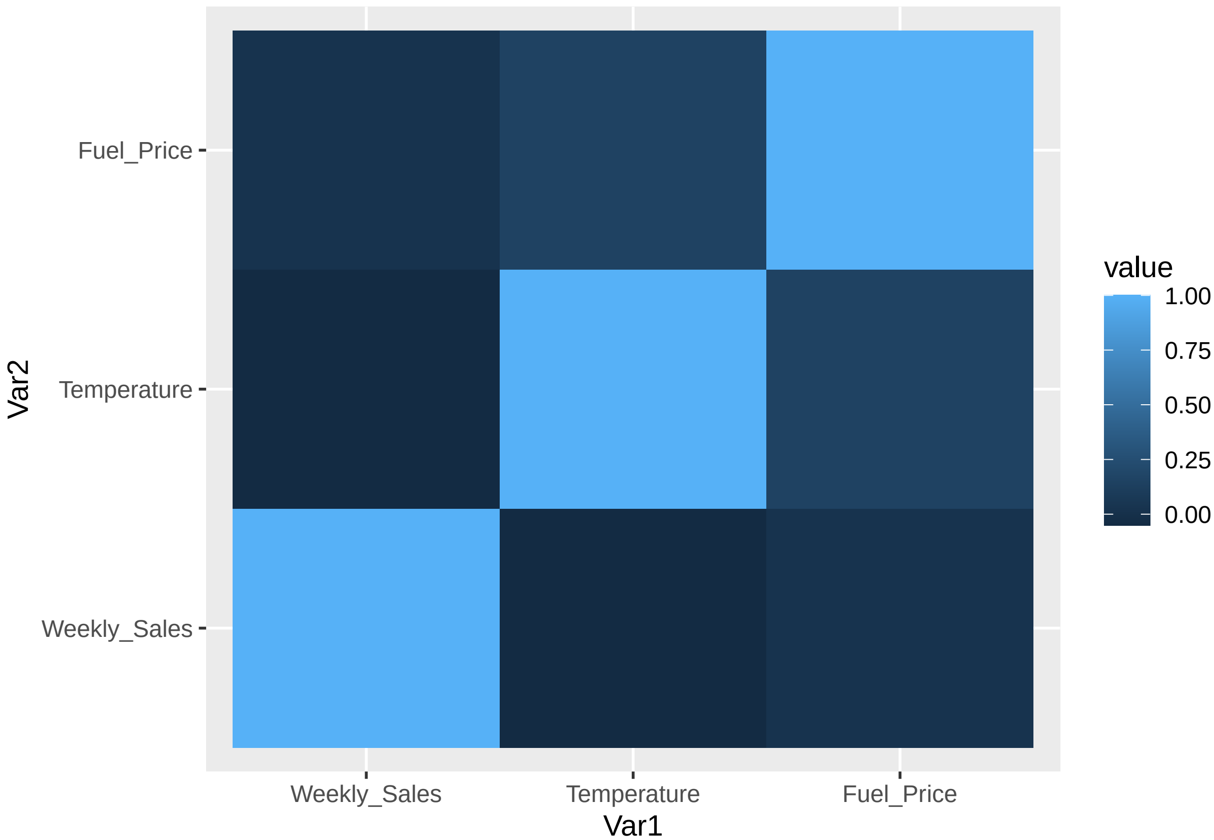
```r
#CORRELATION HEATMAP
install.packages("reshape2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
# reduce the size of correlation matrix
melted_corr_mat <- melt(corr_m)
# head(melted_corr_mat)

# section c questio 2 plotting the correlation heatmap
library(ggplot2)
ggplot(data = melted_corr_mat, aes(x=Var1, y=Var2,
                                   fill=value)) +
  geom_tile()
```




```
####################################33#######333#
#SECTION B (20 Marks)
#creating the two  dataframes
StudentID <- c(101,102,103,104,105,106)
Product <- c("Biology","Math","English","Science","Polical Science","Physics")
df1 <- cbind(StudentID,Product)
head(df1)
```

```
##      StudentID Product
## [1,] "101"     "Biology"
## [2,] "102"     "Math"
## [3,] "103"     "English"
## [4,] "104"     "Science"
## [5,] "105"     "Polical Science"
## [6,] "106"     "Physics"
```

```
#creating the second dataframe
StudentID <- c(102,104,106,107,108)
State <- c("Kuala Lumpur","Johor","Penang","Melaka","Kuala Lumpu")
df2 <- cbind(StudentID,State)
head(df2)
```

```
##      StudentID State
## [1,] "102"     "Kuala Lumpur"
## [2,] "104"     "Johor"
## [3,] "106"     "Penang"
## [4,] "107"     "Melaka"
## [5,] "108"     "Kuala Lumpu"
```

```r
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
library(tidyverse)
#left_join(df1,df2,by="StudentID")
#df3


#3. Write R codes to display only the StudentId and Product which contain
#missing values for State in df2. Show the output of the new dataframe as
#"df4".
#df3 %>%
  #select(StudentID,Product) %>%
  #filter(df3, State == 'NA')



#4. Create "df5" with two variables, StudentId and Marks for 10 students with IDs
#ranging from 101 until 110. Add th
StudentID <- c(101,102,103,104,105,106,107,108,109,110)
Marks <- c(70,90,87,95,93,86,NA,NA,NA,NA)
df5 <- cbind(StudentID,Marks)

head(df5)
```

```
##      StudentID Marks
## [1,]       101    70
## [2,]       102    90
## [3,]       103    87
## [4,]       104    95
## [5,]       105    93
## [6,]       106    86
```

```r
#
#df6
#df7 <- inner_join(df5,df6,by="StudentID")


########################################################33
#SECTION C
#QUESTION ONE

weather <- read_csv("sammyR/weather.csv")
```

```
## Rows: 1461 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): weather
## dbl  (5): year, precipitation, temp_max, temp_min, wind
## date (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
summary(weather)#descriptive statistics
```

```
##       date                 year      precipitation       temp_max
##  Min.   :2012-01-01   Min.   :2012   Min.   : 0.000   Min.   :-1.60
##  1st Qu.:2012-12-31   1st Qu.:2012   1st Qu.: 0.000   1st Qu.:10.60
##  Median :2013-12-31   Median :2013   Median : 0.000   Median :15.60
##  Mean   :2013-12-31   Mean   :2013   Mean   : 3.029   Mean   :16.44
##  3rd Qu.:2014-12-31   3rd Qu.:2014   3rd Qu.: 2.800   3rd Qu.:22.20
##  Max.   :2015-12-31   Max.   :2015   Max.   :55.900   Max.   :35.60
##     temp_min           wind          weather
##  Min.   :-7.100   Min.   :0.400   Length:1461
##  1st Qu.: 4.400   1st Qu.:2.200   Class :character
##  Median : 8.300   Median :3.000   Mode  :character
##  Mean   : 8.235   Mean   :3.241
##  3rd Qu.:12.200   3rd Qu.:4.000
##  Max.   :18.300   Max.   :9.500
```

```r
head(weather,3)#gives the first 3 variables
```

```
## # A tibble: 3 x 7
##   date        year precipitation temp_max temp_min  wind weather
##   <date>     <dbl>         <dbl>    <dbl>    <dbl> <dbl> <chr>
## 1 2012-01-01  2012           0       12.8      5     4.7 drizzle
## 2 2012-01-02  2012          10.9     10.6      2.8   4.5 rain
## 3 2012-01-03  2012           0.8     11.7      7.2   2.3 rain
```

```r
names(weather)#column names ie the variables
```

```
## [1] "date"          "year"          "precipitation" "temp_max"
## [5] "temp_min"      "wind"          "weather"
```

```r
#2. Using the ggplot2 R package, create a correlation heatmap with correlation
#coefficient labels (1 decimal place) to evaluate the relationship between all the
#numerical variables (predictor variables) of weather.
library("dplyr")
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
 fg <- weather %>%
  dplyr::select('precipitation','temp_max','temp_min','wind')
head(fg)
```

```
## # A tibble: 6 x 4
##   precipitation temp_max temp_min  wind
##           <dbl>    <dbl>    <dbl> <dbl>
## 1           0       12.8      5     4.7
## 2          10.9     10.6      2.8   4.5
## 3           0.8     11.7      7.2   2.3
## 4          20.3     12.2      5.6   4.7
## 5           1.3      8.9      2.8   6.1
## 6           2.5      4.4      2.2   2.2
```

```r
install.packages("lattice")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

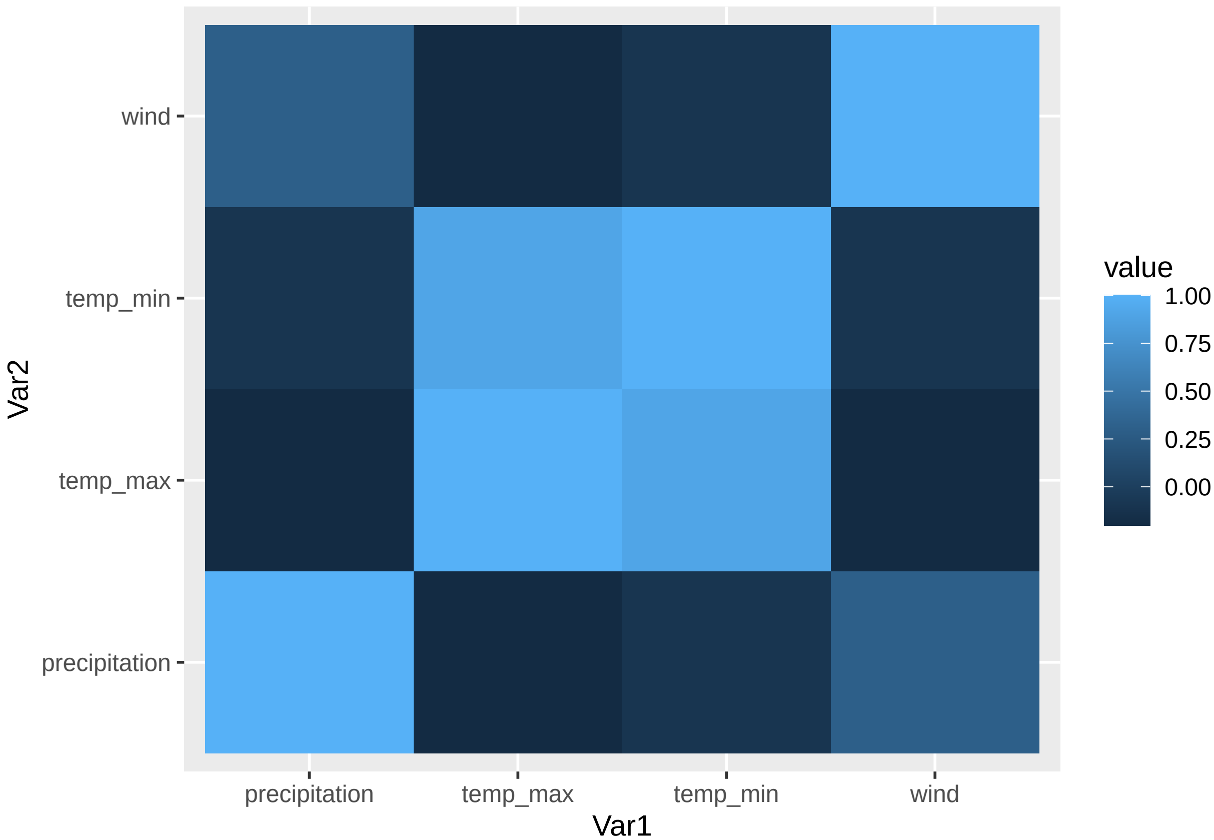
```r
library(lattice)
# rounding to 2 decimal places
corr_mat <- round(cor(fg),1)
head(corr_mat)
```

```
##               precipitation temp_max temp_min wind
## precipitation           1.0     -0.2     -0.1  0.3
## temp_max               -0.2      1.0      0.9 -0.2
## temp_min               -0.1      0.9      1.0 -0.1
## wind                    0.3     -0.2     -0.1  1.0
```

```r
#CORRELATION HEATMAP
# Install and load reshape2 package
install.packages("reshape2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
library(reshape2)
# creating correlation matrix
corr_mat <- round(cor(fg),1)
# reduce the size of correlation matrix
melted_corr_mat <- melt(corr_mat)
# head(melted_corr_mat)
# section c questio 2 plotting the correlation heatmap
library(ggplot2)
ggplot(data = melted_corr_mat, aes(x=Var1, y=Var2,
                                   fill=value)) +
  geom_tile()
```

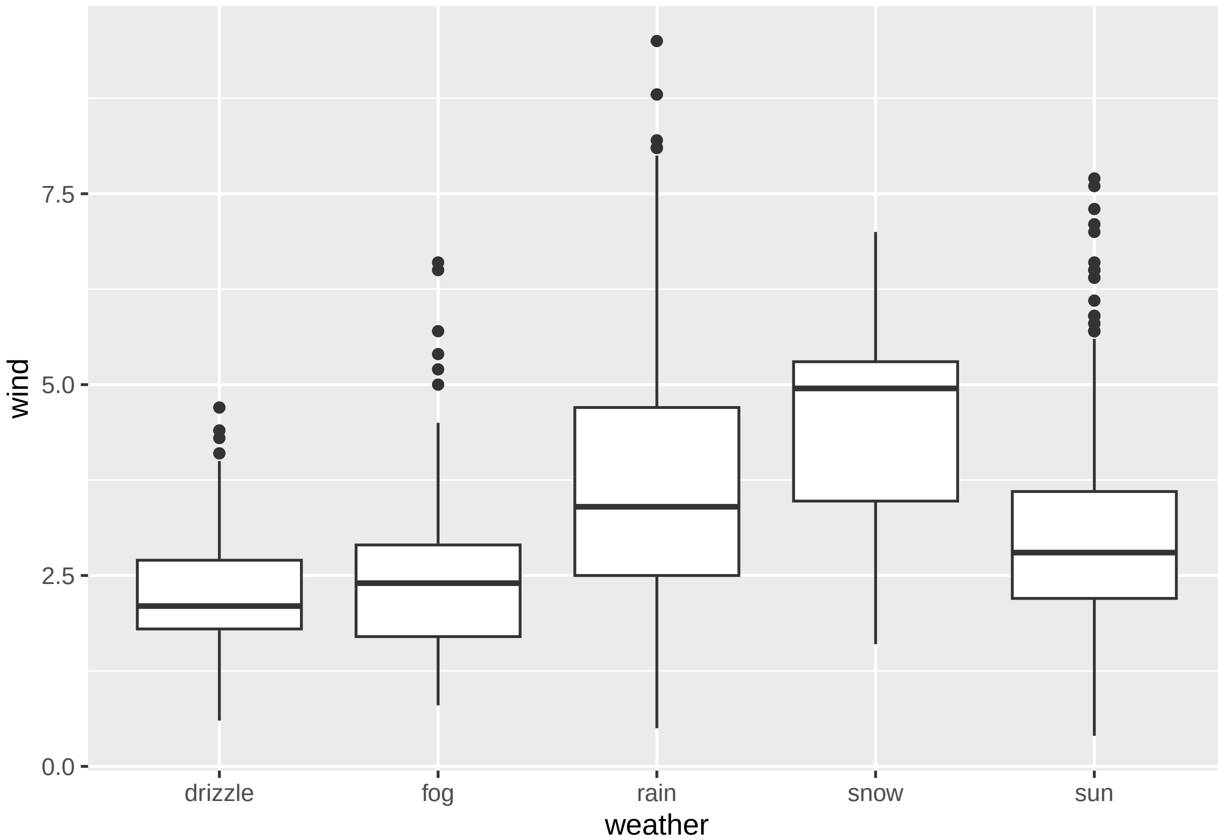



```
weather
```

```
## # A tibble: 1,461 x 7
##    date        year precipitation temp_max temp_min  wind weather
##    <date>     <dbl>         <dbl>    <dbl>    <dbl> <dbl> <chr>
##  1 2012-01-01  2012           0       12.8      5     4.7 drizzle
##  2 2012-01-02  2012          10.9     10.6      2.8   4.5 rain
##  3 2012-01-03  2012           0.8     11.7      7.2   2.3 rain
##  4 2012-01-04  2012          20.3     12.2      5.6   4.7 rain
##  5 2012-01-05  2012           1.3      8.9      2.8   6.1 rain
##  6 2012-01-06  2012           2.5      4.4      2.2   2.2 rain
##  7 2012-01-07  2012           0        7.2      2.8   2.3 rain
##  8 2012-01-08  2012           0       10        2.8   2   sun
##  9 2012-01-09  2012           4.3      9.4      5     3.4 rain
## 10 2012-01-10  2012           1        6.1      0.6   3.4 rain
## # i 1,451 more rows
```

```
#3 From the correlation heatmap above it is evident that there
#is a high correlation between temperature and minimum temperature maximum
#a correlationof 0.9
#you can also observed that there is a low negative correlation between precipitation and the temperatu
#There is a low correlation between wind and temperature minimum this
#correlation is negative
#there is a positive correlation between wind and precipitation correlation
#of 0.3 there is a high correlation between temperature maximum and
#precipitation this correlation is negative
```

```
#question 4


ggplot(data=weather,aes(x=weather ,y=wind) )+
  geom_boxplot()
```




```
#5.  Snow does not contain any outliers




#6, Wind has a high correlation with the  precipitation and this correlation is positive ie 0.3
```