Mini Project- Cold Storage Case Study

Statistical Methods for Decision Making

Table of Contents

1. Project Objective

The objective of the report is to explore the Cold Storage data set (“Cold_Storage_Temp_Data.csv & 01 Cold_Storage_Mar2018.csv”) in R and generate insights about the data set. This exploration report will consists of the following:

- Importing the dataset in R
- Understanding the structure of dataset
- Graphical exploration
- Descriptive statistics
- Insights from the dataset

2. Assumptions

Following assumption we made for this analysis

- The Data Provided to us was not tempered.
- Linearity - Linearity assumes a straight line relationship between each of the two variables.
- Homoscedasticity - Homoscedasticity assumes that data is equally distributed about the regression line.

3. Exploratory Data Analysis Step by Step approach

A Typical Data exploration activity consists of the following steps:

1. Environment Set up and Data Import
2. Variable Identification
3. Univariate Analysis
4. Bi-Variate Analysis
5. Outlier Identification
6. Feature Creation & Exploration

We shall follow these steps in exploring the provided dataset.

3.1 Environment Set up and Data Import

3.1.1 Install necessary Packages and Invoke Libraries

Following are the Libraries are used in the analysis

Package	Library
dplyr	dplyr
ggplot2	ggplot2
tidyverse	tidyverse
DataExplorer	DataExplorer

Code for loading library

```
#Libraries Required
library(tidyverse)
library(dplyr)
library(ggplot2)
library(DataExplorer)
```

Please refer to Appendix A for Source Code.

3.1.2 Set up working Directory

Setting a working directory on starting of the R session makes importing and exporting data files and code files easier. Basically, working directory is the location/ folder on the PC where you have the data, codes etc. related to the project.

Code for setting working directory

```
#Setting the Working Directory
setwd("E:/000GL/000 0Projects/002 Project Cold Storage")
getwd()
```

Please refer to Appendix A for Source Code.

3.1.3 Import and Read the Dataset

The given dataset is in .csv format. Hence, the command 'read.csv' is used for importing the file.

Code for Read the Dataset

```
# Importing Data
## Import the Cold_Storage_Temp_Data.csv
Cold_Storage_Temp = read.csv("02 Cold_Storage_Temp_Data.csv")
Cold_Storage_Temp
```

Please refer to Appendix A for Source Code.

3.2 Variable Identification

Functions is used for variable identifications with there functionality:

- **class(myData):** To identify the class of Data
- **str(myData):** compactly display the (abbreviated) contents of lists.
- **names(myData):** Names of DataFrame variable
- **dim(myData):** Dimensions of Dataframe
- **head(myData):** Display top 6 elements of Variables
- **tail(myData):** Display last 6 elements of variables
- **summary(myData):** Provides an overview of Data
- **plot_missing(myData):** Plot if the variable having any data missing

Code for general Variable Identification

```
#Variable Identification
##Check the Class of Data
class(Cold_Storage_Temp)
```

```
## First Inspection of Dataset using str
str(Cold_Storage_Temp)

## Find the name of variable
names(Cold_Storage_Temp)

## find the dimension of Data
dim(Cold_Storage_Temp)

## find first 6 elements of Data
head(Cold_Storage_Temp)

## find last 5 elements of Data
tail(Cold_Storage_Temp)

## find summary of myData to get Min,median,Mean and Max with First and 3rd quartile.
summary(Cold_Storage_Temp)

## plot the missing value
plot_missing(Cold_Storage_Temp)
```

Please refer to Appendix A for Source Code.

3.2.1 Variable Identification – Inferences

Our Data contain 365 obs. of 4 variables with 3 variables as factors and 1 numerical data.
Column name of our Data are:

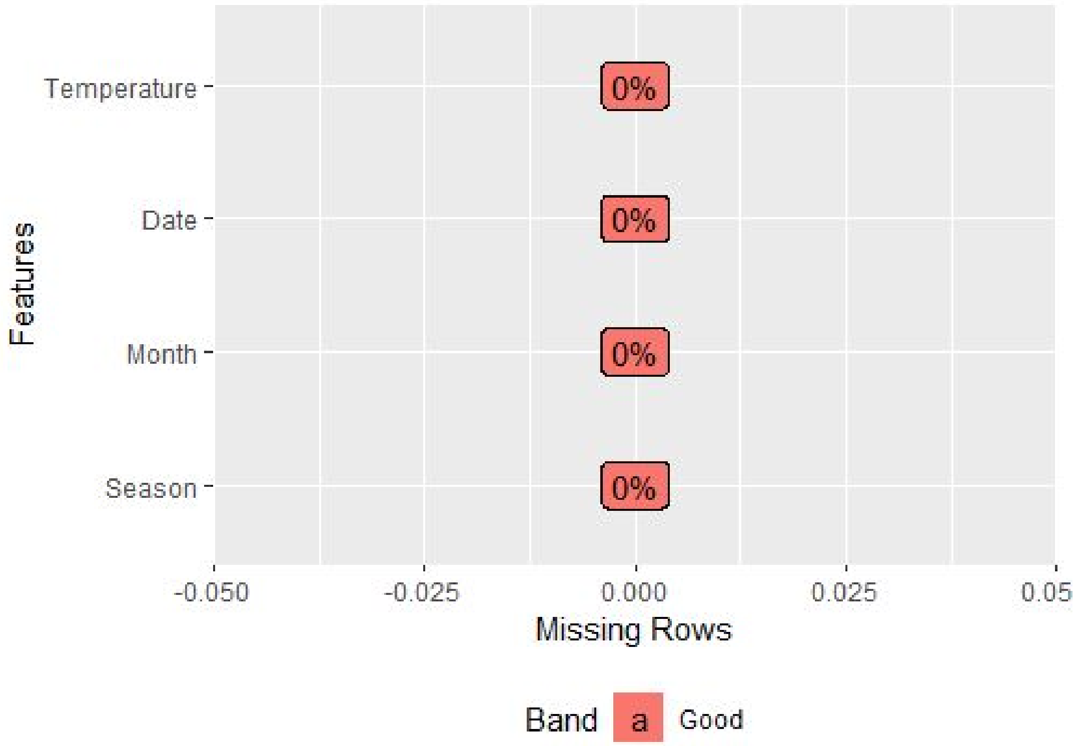
- "Season"
- "Month"
- "Date"
- "Temperature"

We also checked the top 6 and last 6 elements of each variable with command head and tail and summary of data as below.

Command for variable identifications and Output

```
> # General Analysis
> #Variable Identification
> ##Check the Class of Data
> class(Cold_Storage_Temp)
[1] "data.frame"
>
> ## First Inspection of Dataset using str
> str(Cold_Storage_Temp)
'data.frame':	365 obs. of  4 variables:
 $ Season     : Factor w/ 3 levels "Rainy","Summer",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ Month      : Factor w/ 12 levels "Apr","Aug","Dec",..: 5 5 5 5 5 5 5 5 5 5 ...
 $ Date       : Factor w/ 31 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ Temperature: num  2.4 2.3 2.4 2.8 2.5 2.4 2.8 2.3 2.4 2.8 ...
>
```

```
> ## Find the name of variable
> names(Cold_Storage_Temp)
[1] "Season"      "Month"       "Date"        "Temperature"
>
> ## find the dimension of Data
> dim(Cold_Storage_Temp)
[1] 365   4
>
> ## find first 6 elements of Data
> head(Cold_Storage_Temp)
  Season Month Date Temperature
1 Winter   Jan    1         2.4
2 Winter   Jan    2         2.3
3 Winter   Jan    3         2.4
4 Winter   Jan    4         2.8
5 Winter   Jan    5         2.5
6 Winter   Jan    6         2.4
>
> ## find last 5 elements of Data
> tail(Cold_Storage_Temp)
    Season Month Date Temperature
360 Winter   Dec   26         2.7
361 Winter   Dec   27         2.7
362 Winter   Dec   28         2.3
363 Winter   Dec   29         2.6
364 Winter   Dec   30         2.3
365 Winter   Dec   31         2.9
>
> ## find summary of myData to get Min,median,Mean and Max with First and 3rd quartile.
> summary(Cold_Storage_Temp)
    Season         Month          Date      Temperature
 Rainy :122   Aug     : 31   1      : 12   Min.   :1.700
 Summer:120   Dec     : 31   2      : 12   1st Qu.:2.500
 Winter:123   Jan     : 31   3      : 12   Median :2.900
              Jul     : 31   4      : 12   Mean   :2.963
              Mar     : 31   5      : 12   3rd Qu.:3.300
              May     : 31   6      : 12   Max.   :5.000
              (Other):179   (Other):293
>
> ## plot the missing value
> plot_missing(Cold_Storage_Temp)
```
Please refer to Appendix A for Source Code.



(Missing Variable Plot)

3.3 Univariate Analysis

“summary” provides an overview of data for Univariate Analysis
“hist” is used to plot the histogram of numeric variables.
“boxplot” is used to plot the boxplot of numeric variables and also help us to find outliers.
“sd” is used to find the standard deviation of numerical data

Inference:

- *Season*
  3 Season we have with following no. of days
  - Rainy : 122
  - Summer : 120
  - Winter : 123
- *Month*
  12 Months we have with following no. of days
  - Jan : 31
  - Fab : 28
  - Mar : 31
  - Apr : 30
  - May : 31

- Jun : 30
- Jul : 31
- Aug : 31
- Sep : 30
- Oct : 31
- Nov : 30
- Dec : 31

- Temperature has following attributes
  - Min : 1.700
  - 1st Qu. : 2.5
  - Median : 2.9
  - Mean : 2.963
  - 3rd Qu. : 3.3
  - Max. : 5.00
  - Std. Dev : 0.508589

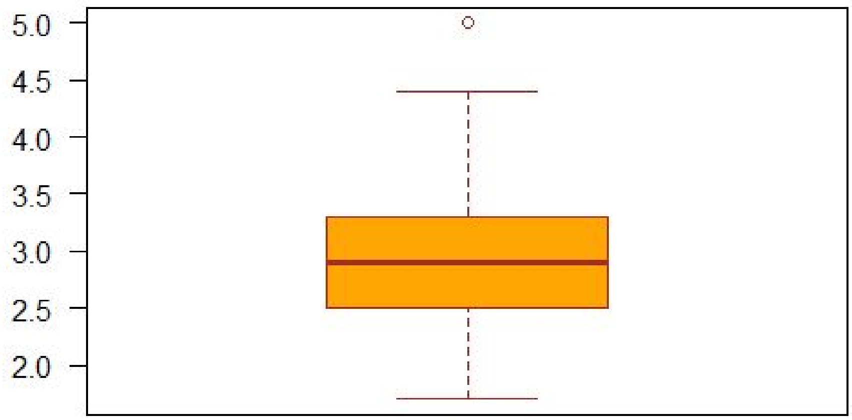
  Box Plot and Histogram of Temperature is as follows

Code for Univariate analysis with output

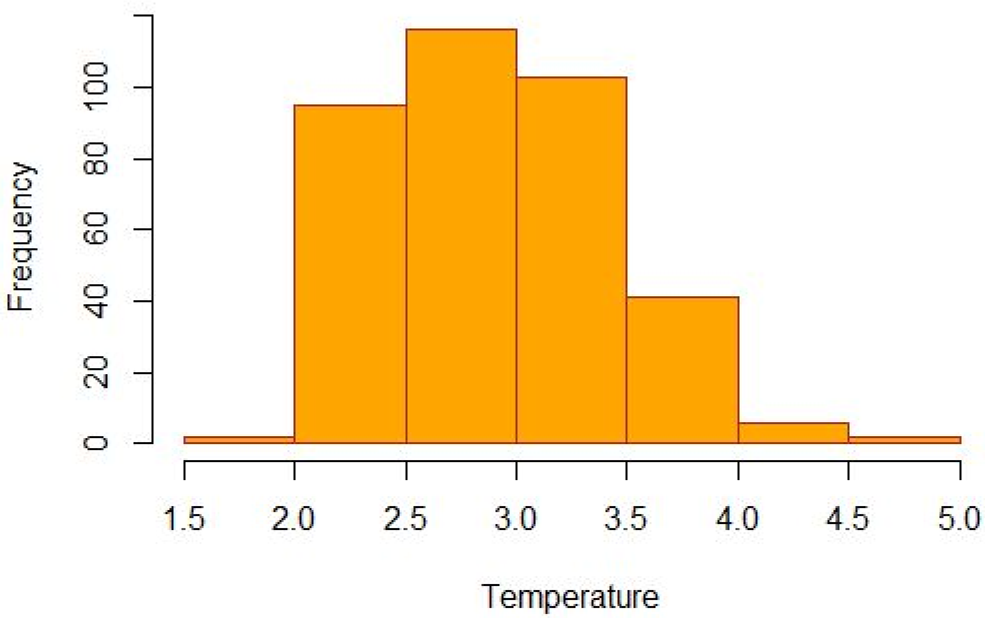
```
> ##Univariate analysis
> ###Season
> summary(Cold_Storage_Temp$Season)
 Rainy Summer Winter 
   122    120    123 
> 
> ###Month
> summary(Cold_Storage_Temp$Month)
Apr Aug Dec Feb Jan Jul Jun Mar May Nov Oct Sep 
 30  31  31  28  31  31  30  31  31  30  31  30 
> 
> ###TEMPERATURE
> summary(Cold_Storage_Temp$Temperature)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.700   2.500   2.900   2.963   3.300   5.000 
> sd(Cold_Storage_Temp$Temperature)
[1] 0.508589
> boxplot(Cold_Storage_Temp$Temperature,las =2
+         ,main = "Box Plot of Temperature"
+         ,col = "orange"
+         ,border = "brown")
> hist(Cold_Storage_Temp$Temperature,
+      main="Histogram for Temperature",
+      xlab="Temperature",
+      border="brown",
+      col="orange",
+      )
```

Please refer to Appendix A for Source Code.

Box Plot of Temperature



Histogram for Temperature



3.4 Bi-Variate Analysis

We uses “groupby” to create table and find relation between seasons and Temperature and Month

Inference :

- *Season wise*:

Season	Days Count	Month Count	Average Temp.
Rainy	122	3.04	4
Summer	120	3.15	4
Winter	123	2.70	4

- *Months wise :*

Month	Days	Average Temp
Jan	31	2.70
feb	28	3.23
Mar	31	3.09
Apr	30	3.13
May	31	3.17
Jun	30	2.97
Jul	31	2.96
Aug	31	3.00
Sep	30	3.23
Oct	31	2.80
Nov	30	2.60
Dec	31	2.70

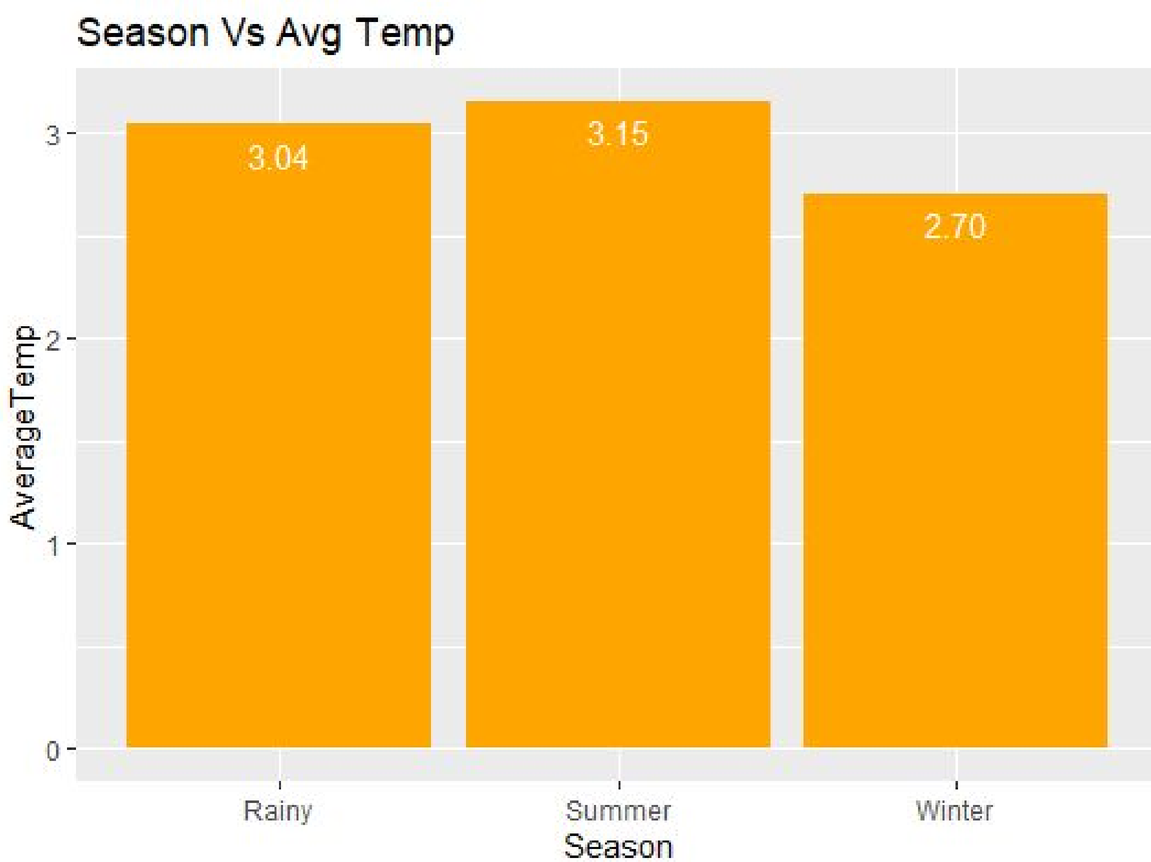
Plots and Output are as below

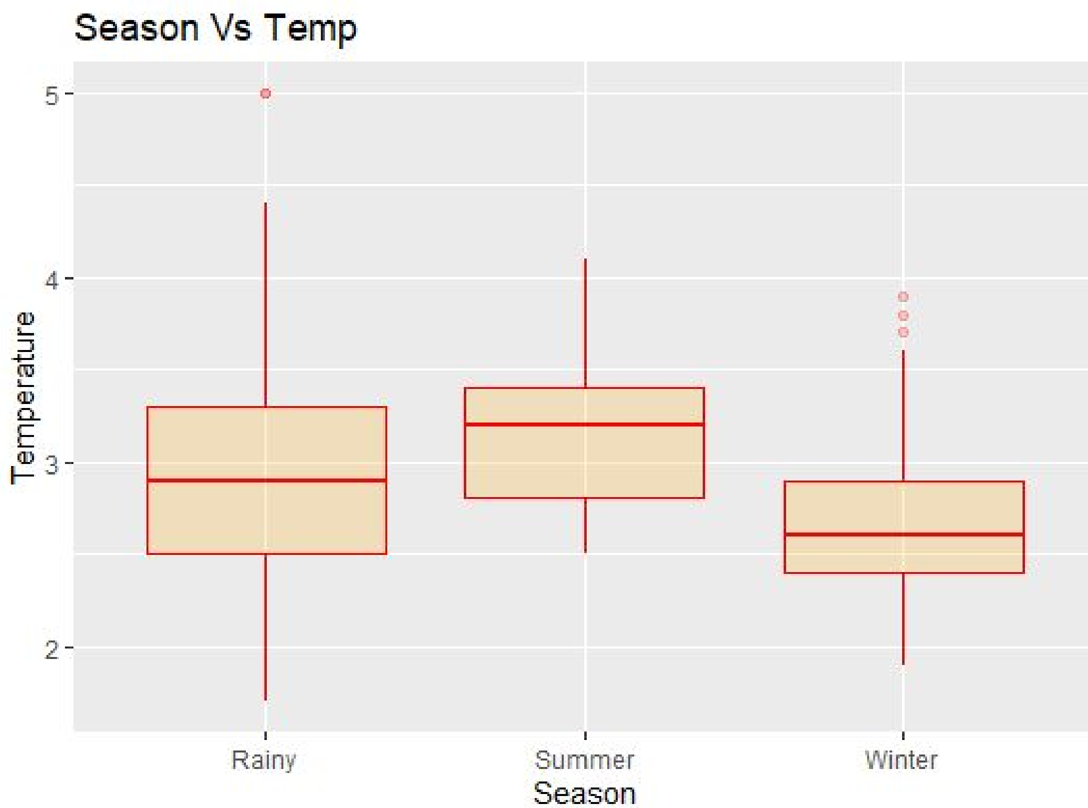
Code for bivariate Analysis

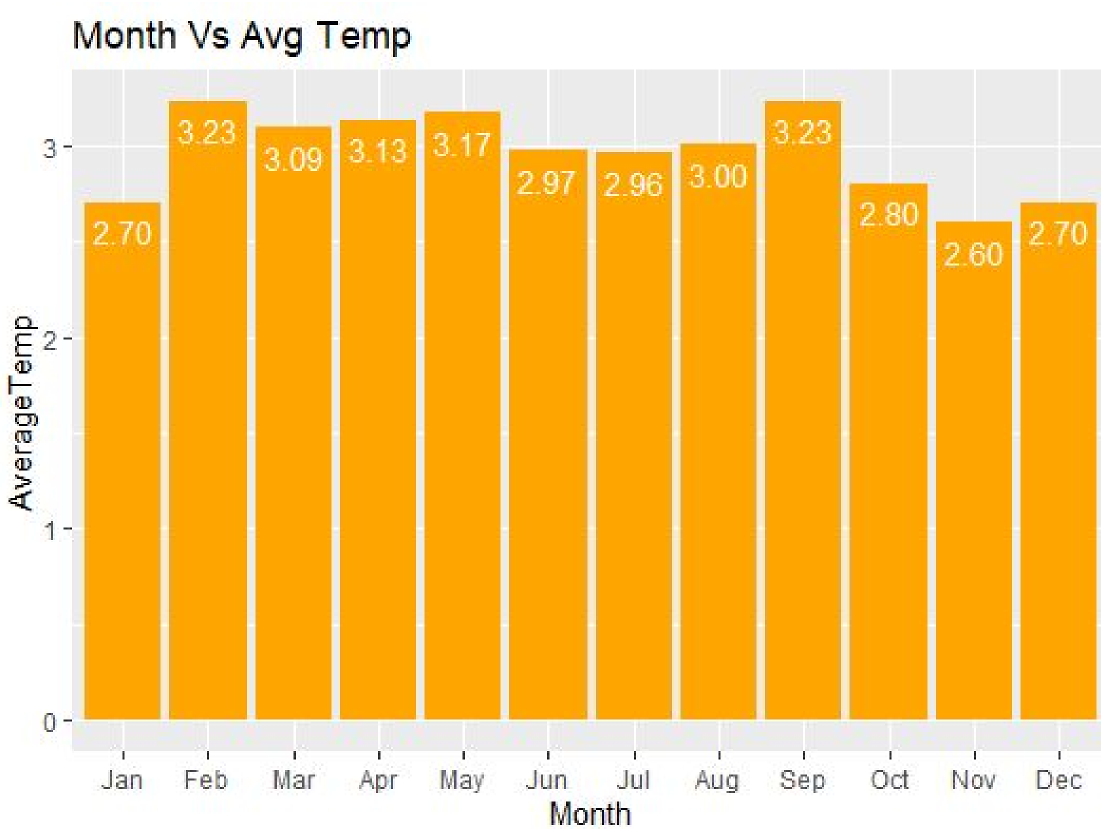
```
> #Bivariate Analysis
> ## Analysis of Cold_Storage_Temp$Season
> ###Table of Season vs DaysCount , Average Temp , Month Count
> Season_Table  = Cold_Storage_Temp %>% group_by(Season) %>%
+                                      summarise(DaysCount = n(),
+                                                AverageTemp = mean(Temperature),
+                                                monthcount = n_distinct(Month))
>
> Season_Table
# A tibble: 3 x 4
  Season DaysCount AverageTemp monthcount
  <fct>      <int>       <dbl>      <int>
1 Rainy        122        3.04          4
2 Summer       120        3.15          4
3 Winter       123        2.70          4
>
> ### Bar Plot of Average Season Temperature
> ggplot(data=Season_Table, aes(x=Season, y=AverageTemp ) ) +
+   geom_bar(stat="identity", fill="orange", width=0.9)+
+   geom_text(aes(label=sprintf("%0.2f", round(AverageTemp, digits = 2))),color="white", vjust=1.6,
size=4) +
+   labs(title = "Season Vs Avg Temp")
>
>
> ### Box Plot of Season wise Temperature
> ggplot(Cold_Storage_Temp, aes(x=Season, y=Temperature)) +
+   geom_boxplot(color="red", fill="orange", alpha=0.2)+
+   labs(title = "Season Vs Temp")
>
>
> ## Analysis of Cold_Storage_Temp$Month
> ### Table of Month vs, Day, Average Temp
> Month_Table = Cold_Storage_Temp %>%   group_by(Month) %>%
+                                      summarise(DaysCount = n(),
+                                                AverageTemp = mean(Temperature))
>
> Month_Table
# A tibble: 12 x 3
   Month DaysCount AverageTemp
   <fct>     <int>       <dbl>
 1 Jan          31        2.70
 2 Feb          28        3.23
 3 Mar          31        3.09
 4 Apr          30        3.13
 5 May          31        3.17
 6 Jun          30        2.97
 7 Jul          31        2.96
 8 Aug          31        3.00
 9 Sep          30        3.23
10 Oct          31        2.80
11 Nov          30        2.60
12 Dec          31        2.70
>
> ### BarPlot of Month vs Average Temp
> ggplot(data=Month_Table, aes(x=Month, y=AverageTemp ) ) +
```

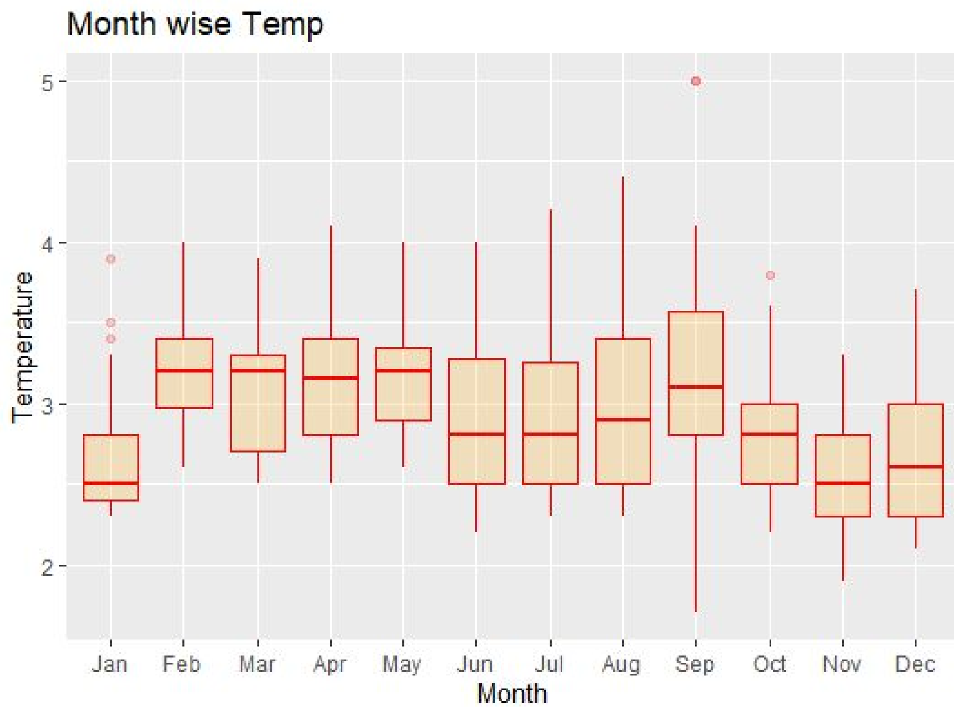
```
+   geom_bar(stat="identity", fill="orange", width=0.9)+
+   geom_text(aes(label=sprintf("%0.2f", round(AverageTemp, digits = 2))),color="white", vjust=1.6,
size=4) +
+   labs(title = "Month Vs Avg Temp")
>
>
> ### BoxPlot of Month wise Temperature
> ggplot(Cold_Storage_Temp, aes(x=Month, y=Temperature)) +
+   geom_boxplot(color="red", fill="orange", alpha=0.2)+
+   labs(title = "Month wise Temp")
>
>
> ### Day wise Temperature of each month
> ggplot(data = Cold_Storage_Temp, aes(x= as.numeric(Date),y=Temperature)) +
+   geom_line(aes(colour=Month))+
+   facet_wrap(~Month) +
+   labs(title = "Month wise Temp", x = "Date")
```

Please refer to Appendix A for Source Code.

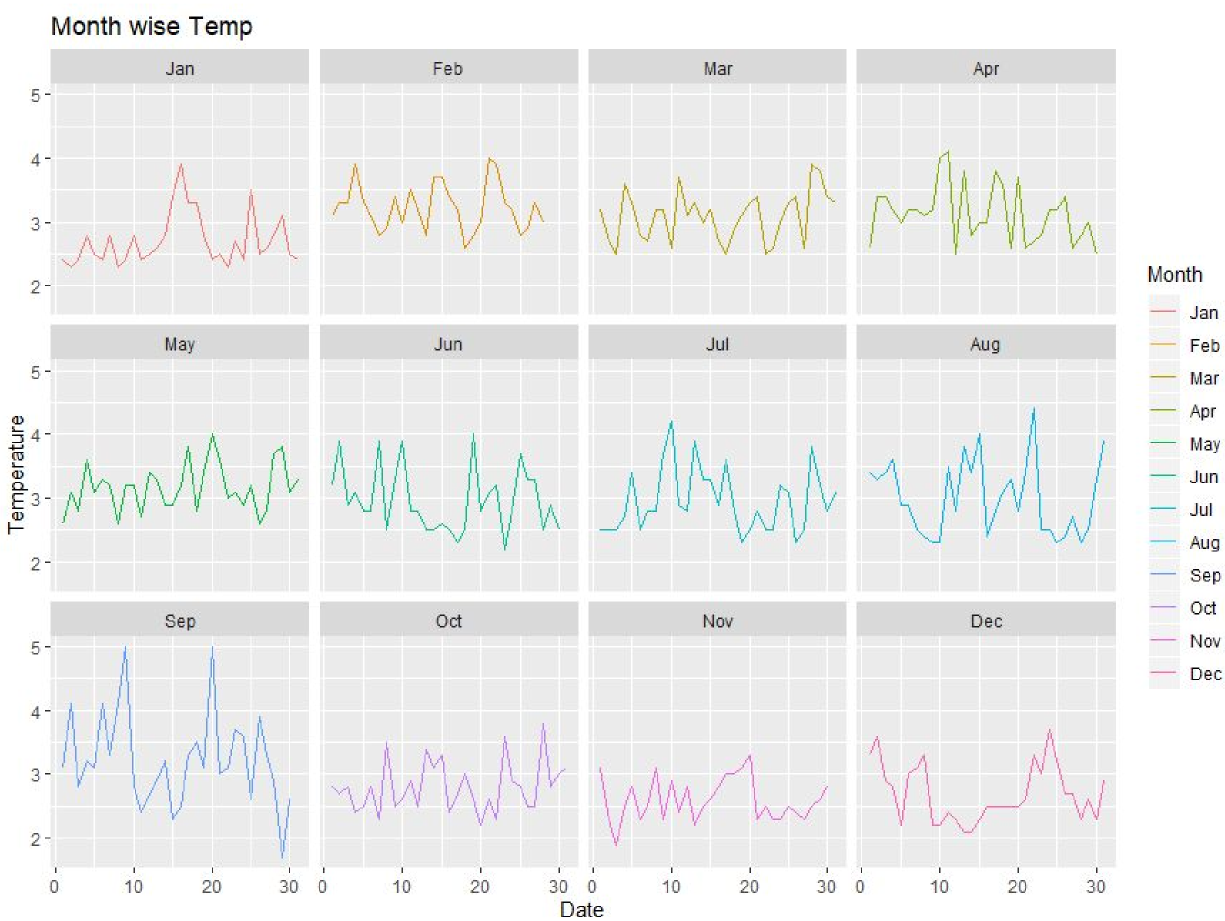



Season Vs Temp
Temperature
5
4
3
2
Rainy
Summer
Winter
Season


Month Vs Avg Temp
AverageTemp
3
2
1
0
2.70
3.23
3.09
3.13
3.17
2.97
2.96
3.00
3.23
2.80
2.60
2.70
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month



Month wise Temp
Temperature
Month
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec



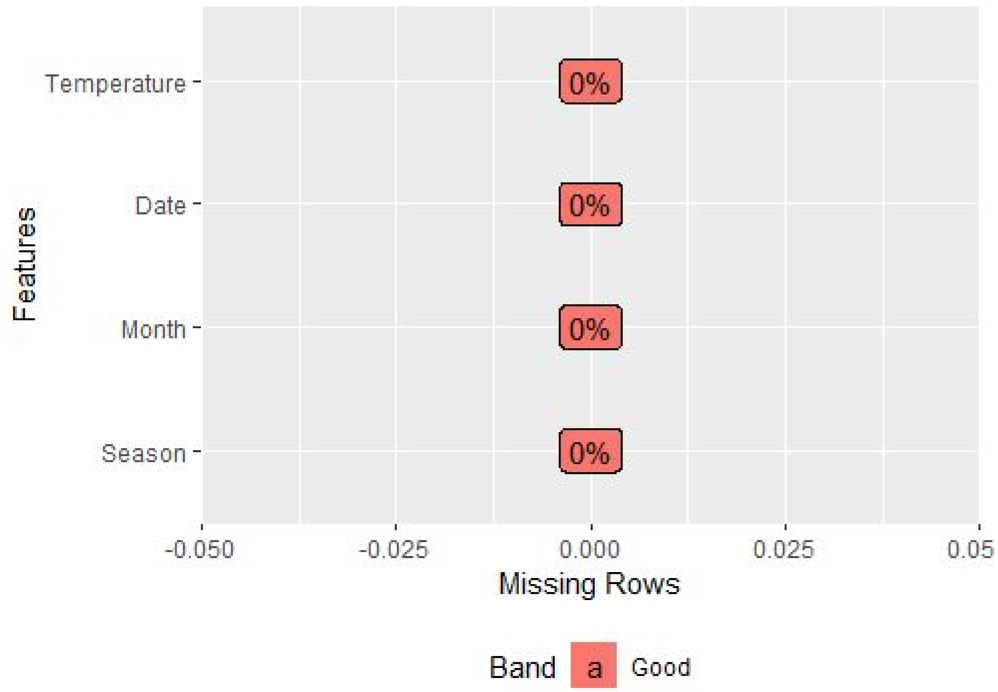
Month wise Temp
Temperature
Date
Month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

3.5 Missing Value Identification

plot_missing(myData) is used to check the missing variable and our data has no missing value

```
> ##plot the missing value
> plot_missing(Cold_Storage_Temp)
```

Please refer to Appendix A for Source Code.

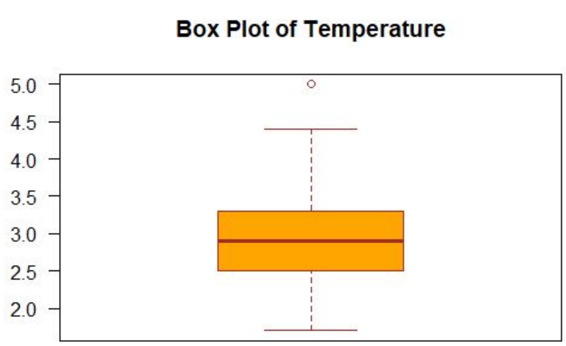


3.6 Outlier Identification

Inference:

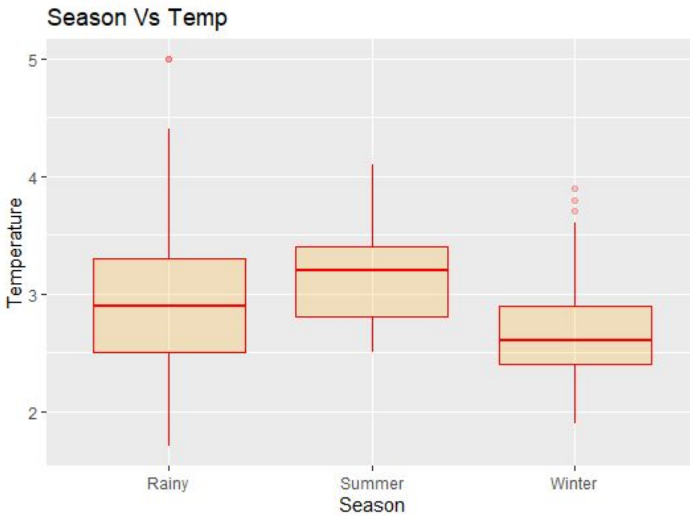
- Overall Temperature Outlier:

  Yes Outlier exist when we consider the Overall Temperature



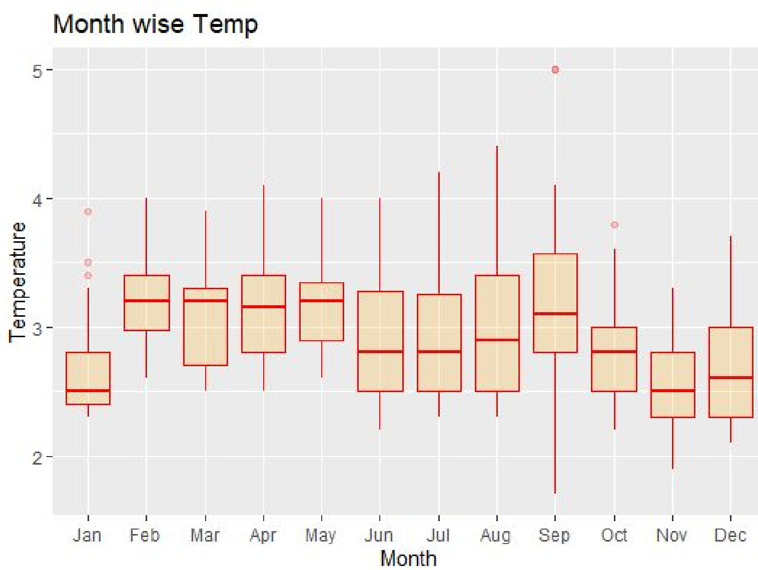
- <u>Season wise Temperature Outlier</u>

  Yes, It exist in season “Rainy” and “Winter” Season



- <u>Month Wise Outlier in Temperature</u>

  Yes, It exist in the month of “Jan” , “Sep”, “Oct”



3.7 Feature Creation

Month_Table , Season_Table are the two table created to get information Month and season wise resp.

4 Conclusion

2 Problem was assigned to us and here is the solution

4.1 Problem 1

4.1.1 Find mean cold storage temperature for Summer, Winter and Rainy Season

Season	Mean Temperature
Rainy	3.04
Summer	3.15
Winter	2.70

Code for finding the mean temperature

```
> mean_temp_of_seasons = Cold_Storage_Temp %>%
+   group_by(Season) %>%
+   summarize(average.Temp = mean(Temperature))
> mean_temp_of_seasons
# A tibble: 3 x 2
  Season average.Temp
  <fct>         <dbl>
1 Rainy          3.04
2 Summer         3.15
3 Winter         2.70
```

Please refer to Appendix A for Source Code.

4.1.2 Find overall mean for the full year

Mean Temperature = 2.96274

Code for finding mean temperature of overall year

```
> Mean_Temp = mean(Cold_Storage_Temp$Temperature)
> Mean_Temp
[1] 2.96274
```

Please refer to Appendix A for Source Code.

4.1.3 Find Standard Deviation for the full year

Std. Dev = 0.508589

Code for finding std dev.of temperature of overall year

```
> SD_of_Temp = sd(Cold_Storage_Temp$Temperature)
> SD_of_Temp
[1] 0.508589
```

Please refer to Appendix A for Source Code.

4.1.4 What is the probability of temperature having fallen below 2 deg C?

Probability of Temp. fallen below 2 deg = 0.29

Code for finding probability of temperature fallen below 2 deg

```
> Prob_for_less_than_2 = pnorm(2, mean=Mean_Temp, sd=SD_of_Temp, lower.tail=TRUE)
> Prob_for_less_than_2
[1] 0.02918146
```

Please refer to Appendix A for Source Code.

4.1.5 What is the probability of temperature having gone above 4 deg C ?

Probability of Temp. having gone above 4 deg C = 0.207

Code for finding probability of temperature having gone above 4 deg C

```
> Prob_for_more_than_4 = pnorm(4, mean=Mean_Temp, sd=SD_of_Temp, lower.tail=FALSE)
> Prob_for_more_than_4
[1] 0.02070077
```

Please refer to Appendix A for Source Code.

4.1.6 What will be the penalty for the AMC Company?

Penalty for the AMC Company = 10% of AMC

Code for finding the penalty for the AMC Company

```
> Probibilty_Temp_Outside_2and4 = Prob_for_less_than_2 + Prob_for_more_than_4
>
> if (Probibilty_Temp_Outside_2and4 <= 0.025) {
+   print("No Penalty")
+ } else if (Probibilty_Temp_Outside_2and4 > 0.025 &&  Probibilty_Temp_Outside_2and4 <= 0.05) {
+   print("Penalty is 10% of the AMC fee")
+ } else
+   { print("Penalty is 25% of the AMC fee")}
[1] "Penalty is 10% of the AMC fee"
```

Please refer to Appendix A for Source Code.

4.2 Problem 2

4.2.1 State the Hypothesis, do the calculation using z test

Hypothesis

H0: Mu ≤ 3.9

H1: Mu > 3.9

Mean_Mar = 3.974286

Mean_Val = 3.9

n = 35

SD_of_Temp = 0.508589

Zval = 0.8641166

Zcritical = 1.281552

Since Zval < Zcritical therefore we fail to reject hypothesis

There isn't enough data to reject the null hypothesis with 90% of confidence .

Pval = 0.8062381

Tempcrit = 4.010171

If mean temp is more than 4.010171 then only we can reject the null hypothesis.

we don't have enough evidence to prove that given sample belong to the population having mean temperature more than 3.9

Code for z test

```
> #  H0 Hypothesis : Mu ≤ 3.9
> # H1 Hypothesis : Mu > 3.9
> # Find Mean of Temperature of sample data
> Mean_Mar = mean(Cold_Storage_Mar$Temperature)
> Mean_Mar
[1] 3.974286
> # Find SD of Sample Temperature
> SD_of_Mar = sd(Cold_Storage_Mar$Temperature)
> SD_of_Mar
[1] 0.159674
> # Mean Value
> Mean_Val = 3.9
> #No. of Observation
> n=35
> #Calculate Z value
> zval = (Mean_Mar - Mean_Val)/(SD_of_Temp/n^0.5)
> zval
[1] 0.8641166
> #Calculate Pvalue
> Pval = pnorm(zval)
> Pval
[1] 0.8062381
> # Find The Z critical
```

```
> zcrtical = qnorm(0.90)
> zcrtical
[1] 1.281552
> # Find the Standard Error
> sd_err = SD_of_Temp/(n^0.5)
> sd_err
[1] 0.08596724
> # Find the Critical Temprature
> Tempcrit = (zcrtical*sd_err)+Mean_Val
> Tempcrit
[1] 4.010171
```

Please refer to Appendix A for Source Code.

4.2.2 State the Hypothesis, do the calculation using t-test

Hypothesis

H0: Mu ≤ 3.9

H1: Mu > 3.9

With t Test we are able to reject the null hypothesis.

That mean we are 90% confident that the sample belong to population having mean greater than 3.9.

t = 2.7524

df = 34

p-value = 0.004711

90 percent confidence interval for the alternate hypothesis is:

3.939011 - Infinity

There is sufficient evidence that the mean Temperature of population is more than 3.9

Code for t test

```
> t.test(Cold_Storage_Mar$Temperature,mu=3.9,alternative ="greater",conf.level = 0.9)

        One Sample t-test

data:  Cold_Storage_Mar$Temperature
t = 2.7524, df = 34, p-value = 0.004711
alternative hypothesis: true mean is greater than 3.9
90 percent confidence interval:
 3.939011      Inf
sample estimates:
mean of x
 3.974286
```

Please refer to Appendix A for Source Code.

4.2.3 Give your inference after doing both the tests.

Via Z test:

Since **Zval < Zcritical** therefore we fail to reject hypothesis

There isn't enough data to reject the null hypothesis with 90% of confidence .

Inference via z test.

we don't have enough evidence to prove that given sample belong to the population having mean temperature more than 3.9

There is no need for some corrective action in the Cold Storage Plant

The problem might be from procurement side

Via T test:

With **p-value = 0.004711** in t Test we are able to reject the null hypothesis.

That mean we are 90% confident that the sample belong to population having mean greater than 3.9.

Inference via z test.

There is sufficient evidence that the mean Temperature of population is more than 3.9

There is need for some corrective action in the Cold Storage Plant

Inference :

Since In Mar 2018, Cold Storage started getting complaints from their Clients therefore via T Test we have sufficient evidence that the mean Temperature is going more than 3.9

There is need for some corrective action in the Cold Storage Plant.

5 Appendix A – Source Code

```
> #Libraries Required
> library(tidyverse)
> library(dplyr)
> library(ggplot2)
> library(DataExplorer)
> #Setting the Working Directory
> setwd("E:/000GL/000 0Projects/002 Project Cold Storage")
> getwd()
[1] "E:/000GL/000 0Projects/002 Project Cold Storage"
> # Importing Data
> ## Import the Cold_Storage_Temp_Data.csv
> Cold_Storage_Temp = read.csv("02 Cold_Storage_Temp_Data.csv")
> Cold_Storage_Temp$Date = as.factor(Cold_Storage_Temp$Date)
> Cold_Storage_Temp$Month = factor(Cold_Storage_Temp$Month  ,levels = c("Jan", "Feb","Mar","Apr","May",
"Jun", "Jul", "Aug","Sep", "Oct","Nov","Dec"))
> # General Analysis
> #Variable Identification
> ##Check the Class of Data
> class(Cold_Storage_Temp)
[1] "data.frame"
> ## First Inspection of Dataset using str
> str(Cold_Storage_Temp)
'data.frame':	365 obs. of  4 variables:
 $ Season     : Factor w/ 3 levels "Rainy","Summer",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ Month      : Factor w/ 12 levels "Jan","Feb","Mar",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Date       : Factor w/ 31 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ Temperature: num  2.4 2.3 2.4 2.8 2.5 2.4 2.8 2.3 2.4 2.8 ...
> ## Find the name of variable
> names(Cold_Storage_Temp)
[1] "Season"      "Month"       "Date"        "Temperature"
> ## find the dimension of Data
> dim(Cold_Storage_Temp)
[1] 365   4
> ## find first 6 elements of Data
> head(Cold_Storage_Temp)
  Season Month Date Temperature
1 Winter   Jan    1         2.4
2 Winter   Jan    2         2.3
3 Winter   Jan    3         2.4
4 Winter   Jan    4         2.8
5 Winter   Jan    5         2.5
6 Winter   Jan    6         2.4
> ## find last 5 elements of Data
> tail(Cold_Storage_Temp)
    Season Month Date Temperature
360 Winter   Dec   26         2.7
361 Winter   Dec   27         2.7
362 Winter   Dec   28         2.3
363 Winter   Dec   29         2.6
364 Winter   Dec   30         2.3
365 Winter   Dec   31         2.9
> ## find summary of myData to get Min,median,Mean and Max with First and 3rd quartile.
> summary(Cold_Storage_Temp)
```

```
    Season          Month            Date        Temperature
 Rainy :122    Jan     : 31    1       : 12    Min.   :1.700
 Summer:120    Mar     : 31    2       : 12    1st Qu.:2.500
 Winter:123    May     : 31    3       : 12    Median :2.900
               Jul     : 31    4       : 12    Mean   :2.963
               Aug     : 31    5       : 12    3rd Qu.:3.300
               Oct     : 31    6       : 12    Max.   :5.000
               (Other):179    (Other):293
> ## plot the missing value
> plot_missing(Cold_Storage_Temp)
> #Univarient analysis
> ##Season
> summary(Cold_Storage_Temp$Season)
 Rainy Summer Winter
   122    120    123
> ##Month
> summary(Cold_Storage_Temp$Month)
Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
 31  28  31  30  31  30  31  31  30  31  30  31
> ##TEMPERATURE
> summary(Cold_Storage_Temp$Temperature)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.700   2.500   2.900   2.963   3.300   5.000
> sd(Cold_Storage_Temp$Temperature)
[1] 0.508589
> boxplot(Cold_Storage_Temp$Temperature,las =2
+         ,main = "Box Plot of Temperature"
+         ,col = "orange"
+         ,border = "brown")
> hist(Cold_Storage_Temp$Temperature,
+      main="Histogram for Temperature",
+      xlab="Temperature",
+      border="brown",
+      col="orange",
+      )
> #Bivariate Analysis
> ## Analysis of Cold_Storage_Temp$Season
> ###Table of Season vs DaysCount , Aveerage Temp , Month Count
> Season_Table  = Cold_Storage_Temp %>% group_by(Season) %>%
+                                      summarise(DaysCount = n(),
+                                                AverageTemp = mean(Temperature),
+                                                monthcount = n_distinct(Month))
> Season_Table
# A tibble: 3 x 4
  Season DaysCount AverageTemp monthcount
  <fct>      <int>       <dbl>      <int>
1 Rainy        122        3.04          4
2 Summer       120        3.15          4
3 Winter       123        2.70          4
> ### Bar Plot of Average Season Temperature
> ggplot(data=Season_Table, aes(x=Season, y=AverageTemp ) ) +
+   geom_bar(stat="identity", fill="orange", width=0.9)+
+   geom_text(aes(label=sprintf("%0.2f", round(AverageTemp, digits = 2))),color="white", vjust=1.6,
size=4) +
+   labs(title = "Season Vs Avg Temp")
> ### Box Plot of Season wise Temperature
```

```
> ggplot(Cold_Storage_Temp, aes(x=Season, y=Temperature)) +
+   geom_boxplot(color="red", fill="orange", alpha=0.2)+
+   labs(title = "Season Vs Temp")
> ## Analysis of Cold_Storage_Temp$Month
> ### Table of Month vs, Day, Average Temp
> Month_Table = Cold_Storage_Temp %>%   group_by(Month) %>%
+                                       summarise(DaysCount = n(),
+                                                 AverageTemp = mean(Temperature))
> Month_Table
# A tibble: 12 x 3
   Month DaysCount AverageTemp
   <fct>     <int>       <dbl>
 1 Jan          31        2.70
 2 Feb          28        3.23
 3 Mar          31        3.09
 4 Apr          30        3.13
 5 May          31        3.17
 6 Jun          30        2.97
 7 Jul          31        2.96
 8 Aug          31        3.00
 9 Sep          30        3.23
10 Oct          31        2.80
11 Nov          30        2.60
12 Dec          31        2.70
> ### BarPlot of Month vs Average Temp
> ggplot(data=Month_Table, aes(x=Month, y=AverageTemp ) ) +
+   geom_bar(stat="identity", fill="orange", width=0.9)+
+   geom_text(aes(label=sprintf("%0.2f", round(AverageTemp, digits = 2))),color="white", vjust=1.6,
size=4) +
+   labs(title = "Month Vs Avg Temp")
> ### BoxPlot of Month wise Temperature
> ggplot(Cold_Storage_Temp, aes(x=Month, y=Temperature)) +
+   geom_boxplot(color="red", fill="orange", alpha=0.2)+
+   labs(title = "Month wise Temp")
> ### Day wise Temperature of each month
> ggplot(data = Cold_Storage_Temp, aes(x= as.numeric(Date),y=Temperature)) +
+   geom_line(aes(colour=Month))+
+   facet_wrap(~Month) +
+   labs(title = "Month wise Temp", x = "Date")
> #Problem 1
> ## Q1. Mean cold storage temperature for Summer, Winter and Rainy Season
> mean_temp_of_seasons = Cold_Storage_Temp %>%
+   group_by(Season) %>%
+   summarize(average.Temp = mean(Temperature))
> mean_temp_of_seasons
# A tibble: 3 x 2
  Season average.Temp
  <fct>         <dbl>
1 Rainy          3.04
2 Summer         3.15
3 Winter         2.70
> ## Q2.overall mean for the full year
> Mean_Temp = mean(Cold_Storage_Temp$Temperature)
> Mean_Temp
[1] 2.96274
> ## Q3. Standard Deviation for the full year
```

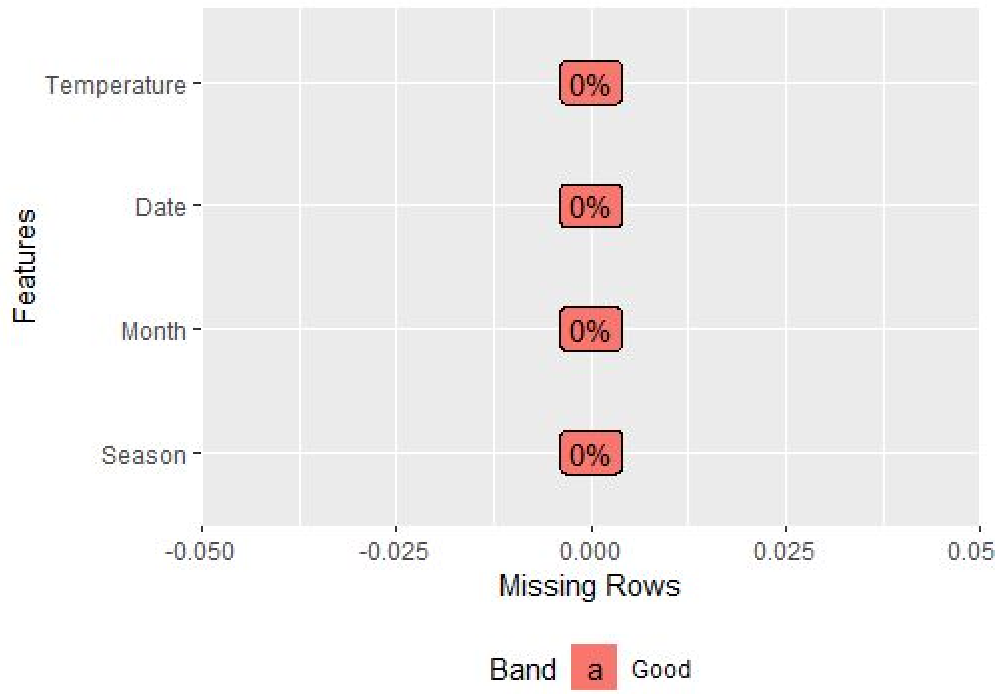
```
> SD_of_Temp = sd(Cold_Storage_Temp$Temperature)
> SD_of_Temp
[1] 0.508589
> ## Q4. probability of temperature having fallen below 2 deg C
> Prob_for_less_than_2 = pnorm(2, mean=Mean_Temp, sd=SD_of_Temp, lower.tail=TRUE)
> Prob_for_less_than_2
[1] 0.02918146
> ## Q5. probability of temperature having gone above 4 deg C
> Prob_for_more_than_4 = pnorm(4, mean=Mean_Temp, sd=SD_of_Temp, lower.tail=FALSE)
> Prob_for_more_than_4
[1] 0.02070077
> ## Q6. penalty for the AMC Company
> Probibilty_Temp_Outside_2and4 = Prob_for_less_than_2 + Prob_for_more_than_4
> if (Probibilty_Temp_Outside_2and4 <= 0.025) {
+   print("No Penalty")
+ } else if (Probibilty_Temp_Outside_2and4 > 0.025 &&  Probibilty_Temp_Outside_2and4 <= 0.05) {
+   print("Penalty is 10% of the AMC fee")
+ } else{
+   print("Penalty is 25% of the AMC fee")
+ }
[1] "Penalty is 10% of the AMC fee"
> #H0 Hypothsis : Mu ??? 3.9
> #H1 Hypothsis : Mu > 3.9
> #Read the "01 Cold_Storage_Mar2018.csv" file
> Cold_Storage_Mar = read.csv("01 Cold_Storage_Mar2018.csv")
> Cold_Storage_Mar
   Season Month Date Temperature
1  Summer   Feb   11         4.0
2  Summer   Feb   12         3.9
3  Summer   Feb   13         3.9
4  Summer   Feb   14         4.0
5  Summer   Feb   15         3.8
6  Summer   Feb   16         4.0
7  Summer   Feb   17         4.1
8  Summer   Feb   18         4.0
9  Summer   Feb   19         3.8
10 Summer   Feb   20         3.9
11 Summer   Feb   21         3.9
12 Summer   Feb   22         4.6
13 Summer   Feb   23         4.1
14 Summer   Feb   24         4.1
15 Summer   Feb   25         3.9
16 Summer   Feb   26         3.8
17 Summer   Feb   27         3.8
18 Summer   Feb   28         3.9
19 Summer   Mar    1         3.9
20 Summer   Mar    2         3.9
21 Summer   Mar    3         3.9
22 Summer   Mar    4         4.1
23 Summer   Mar    5         3.9
24 Summer   Mar    6         3.9
25 Summer   Mar    7         4.1
26 Summer   Mar    8         4.0
27 Summer   Mar    9         4.1
28 Summer   Mar   10         3.9
29 Summer   Mar   11         4.1
```

```
30 Summer   Mar   12         3.8
31 Summer   Mar   13         4.2
32 Summer   Mar   14         4.2
33 Summer   Mar   15         3.8
34 Summer   Mar   16         3.9
35 Summer   Mar   17         3.9
> summary(Cold_Storage_Mar)
    Season    Month         Date         Temperature
 Summer:35   Feb:18   Min.   : 1.0   Min.   :3.800
             Mar:17   1st Qu.: 9.5   1st Qu.:3.900
                      Median :14.0   Median :3.900
                      Mean   :14.4   Mean   :3.974
                      3rd Qu.:19.5   3rd Qu.:4.100
                      Max.   :28.0   Max.   :4.600
> Cold_Storage_Mar = read.csv("01 Cold_Storage_Mar2018.csv")
> #  H0 Hypothsis : Mu ??? 3.9
> # H1 Hypothsis : Mu > 3.9
> # Find Mean of Temperature of sample data
> Mean_Mar = mean(Cold_Storage_Mar$Temperature)
> Mean_Mar
[1] 3.974286
> # Find SD of Sample Temperature
> SD_of_Mar = sd(Cold_Storage_Mar$Temperature)
> SD_of_Mar
[1] 0.159674
> # Mean Value
> Mean_Val = 3.9
> #No. of Observation
> n=35
> #Calculate Z value
> zval = (Mean_Mar - Mean_Val)/(SD_of_Temp/n^0.5)
> zval
[1] 0.8641166
> #Calculate Pvalue
> Pval = pnorm(zval)
> Pval
[1] 0.8062381
> # Find The Z critical
> zcrtical = qnorm(0.90)
> zcrtical
[1] 1.281552
> # Find the Standard Error
> sd_err = SD_of_Temp/(n^0.5)
> sd_err
[1] 0.08596724
> # Find the Critical Temprature
> Tempcrit = (zcrtical*sd_err)+Mean_Val
> Tempcrit
[1] 4.010171
> ##Q2
> t.test(Cold_Storage_Mar$Temperature,mu=3.9,alternative ="greater",conf.level = 0.9)

        One Sample t-test

data:  Cold_Storage_Mar$Temperature
t = 2.7524, df = 34, p-value = 0.004711
```

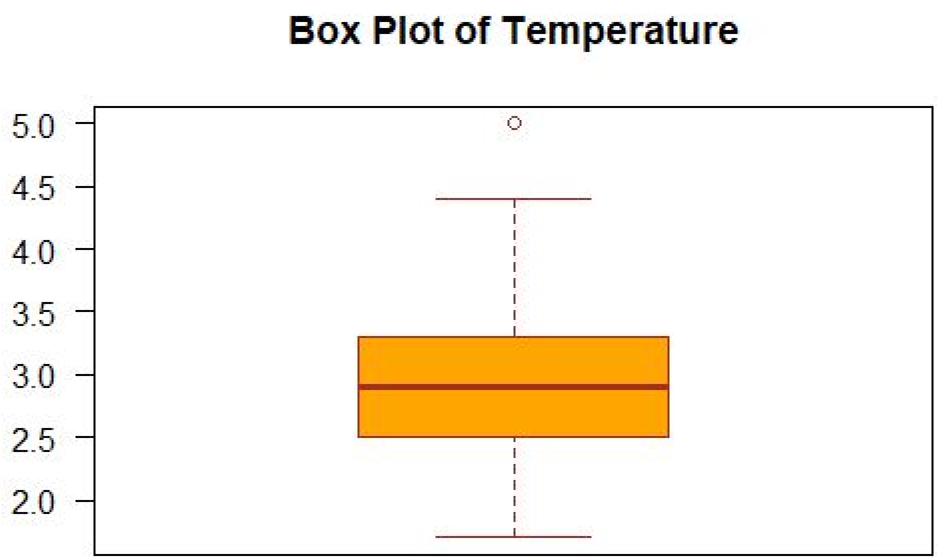
```
alternative hypothesis: true mean is greater than 3.9
90 percent confidence interval:
 3.939011      Inf
sample estimates:
mean of x
 3.974286
```

6 Appendix B – Graphs and Plot

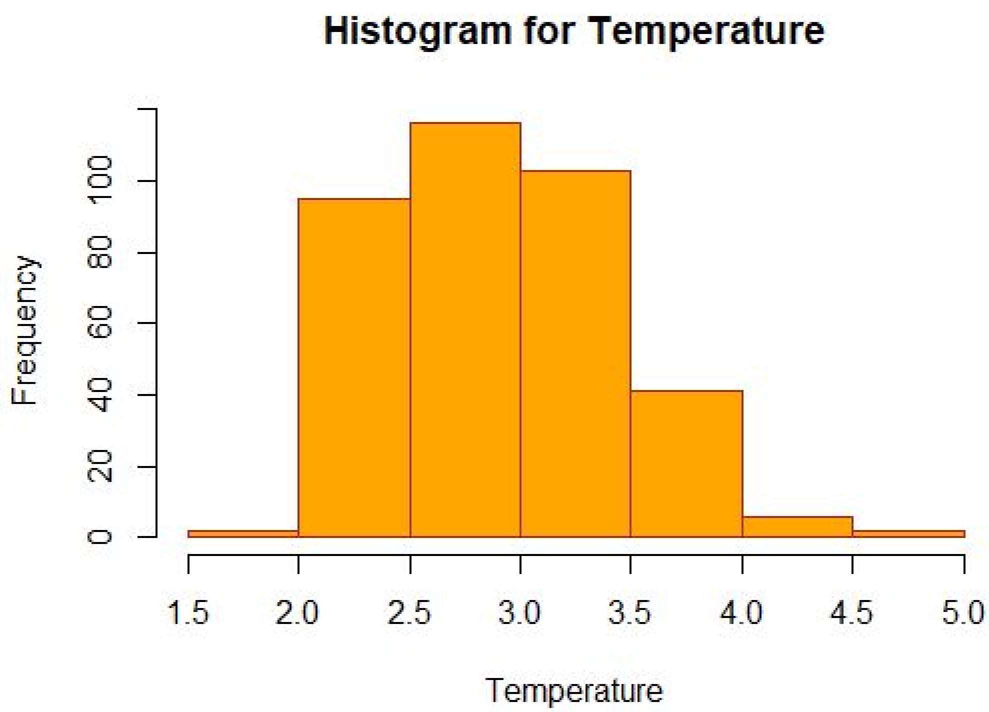
6.1 Missing Variable Plot



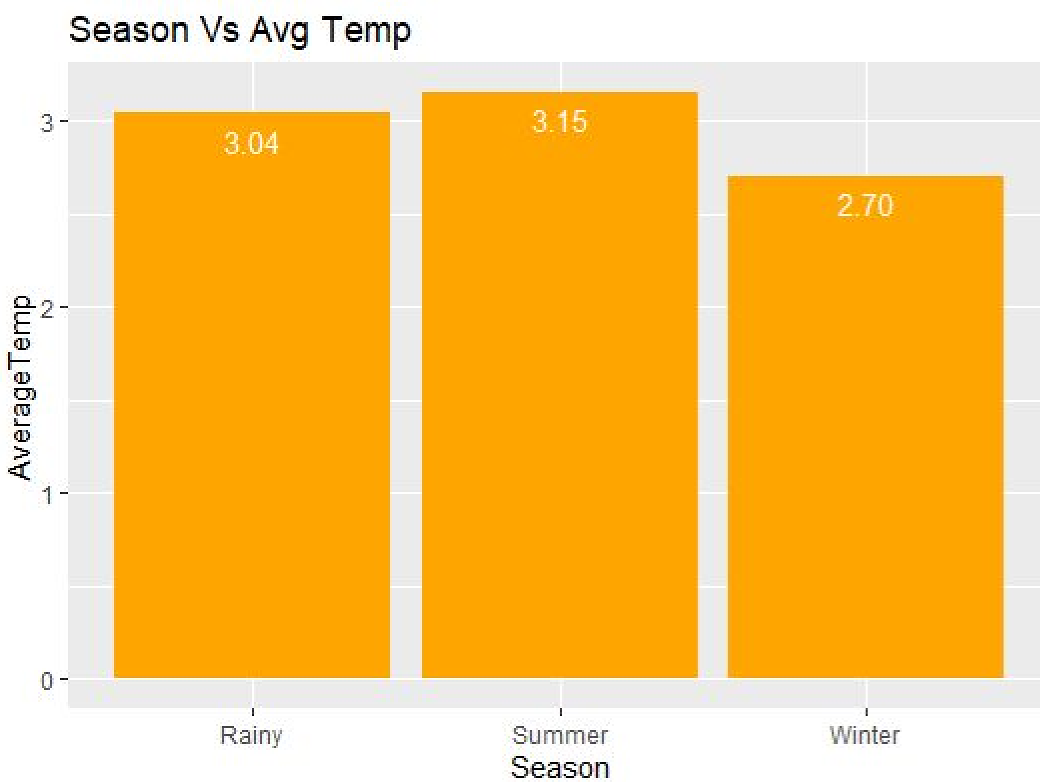
6.2 Box of Temperature (Annual)



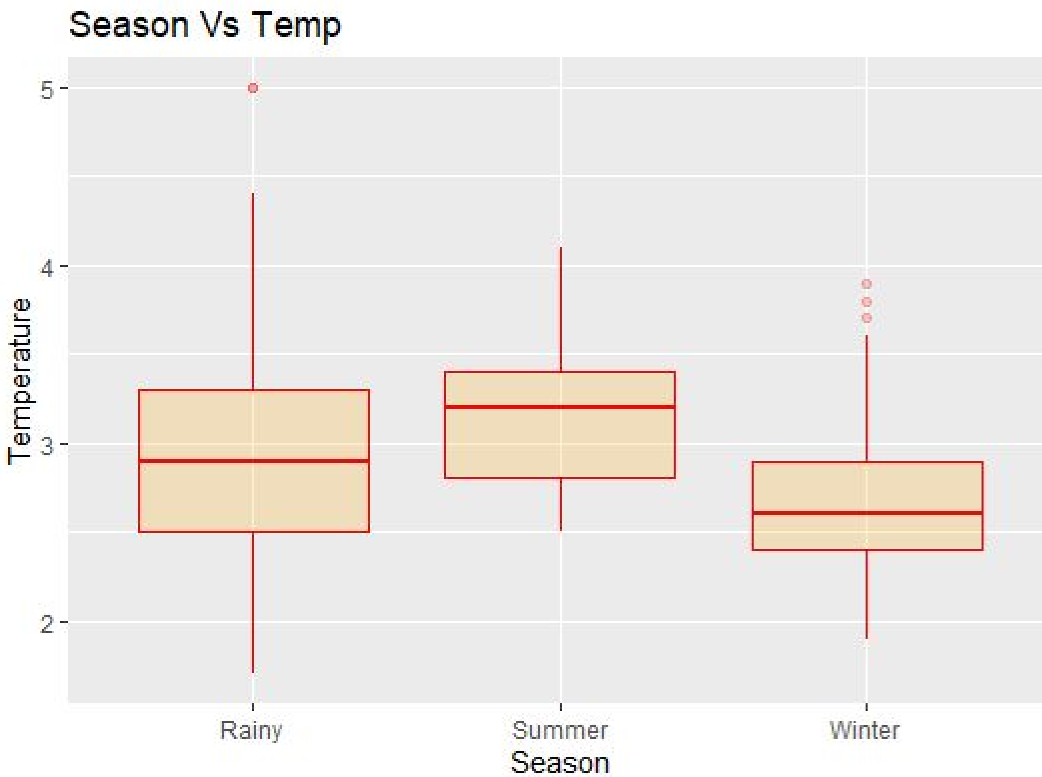
6.3 Histogram of Temperature



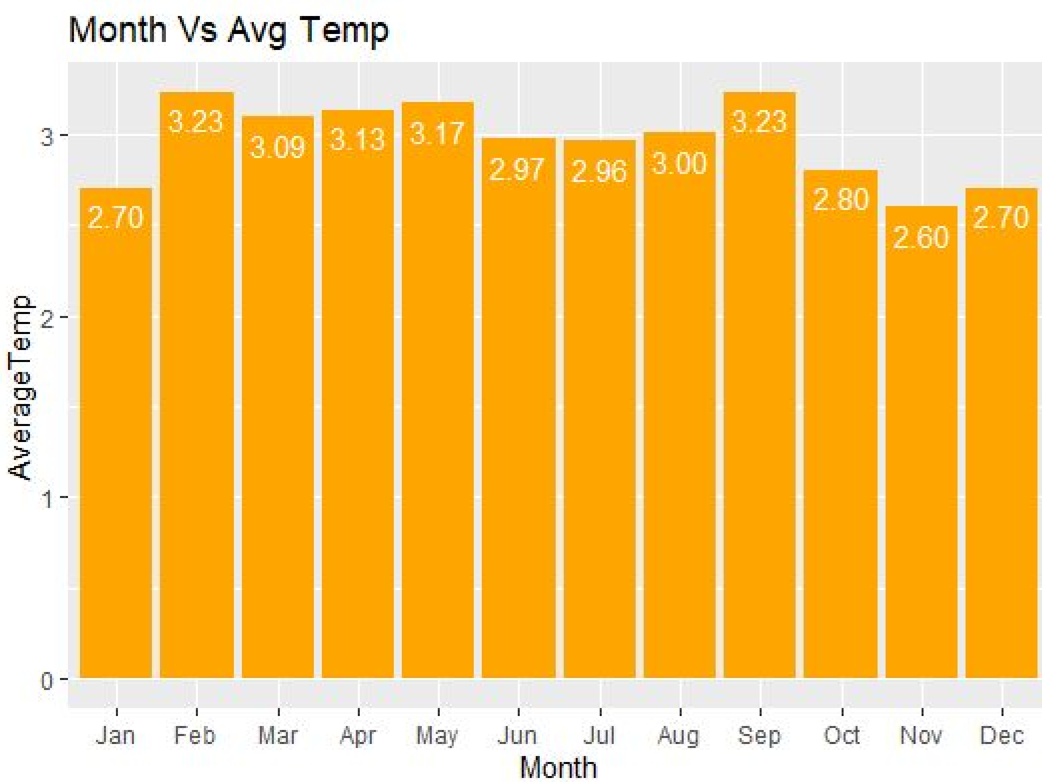
6.4 Average Season Temperature



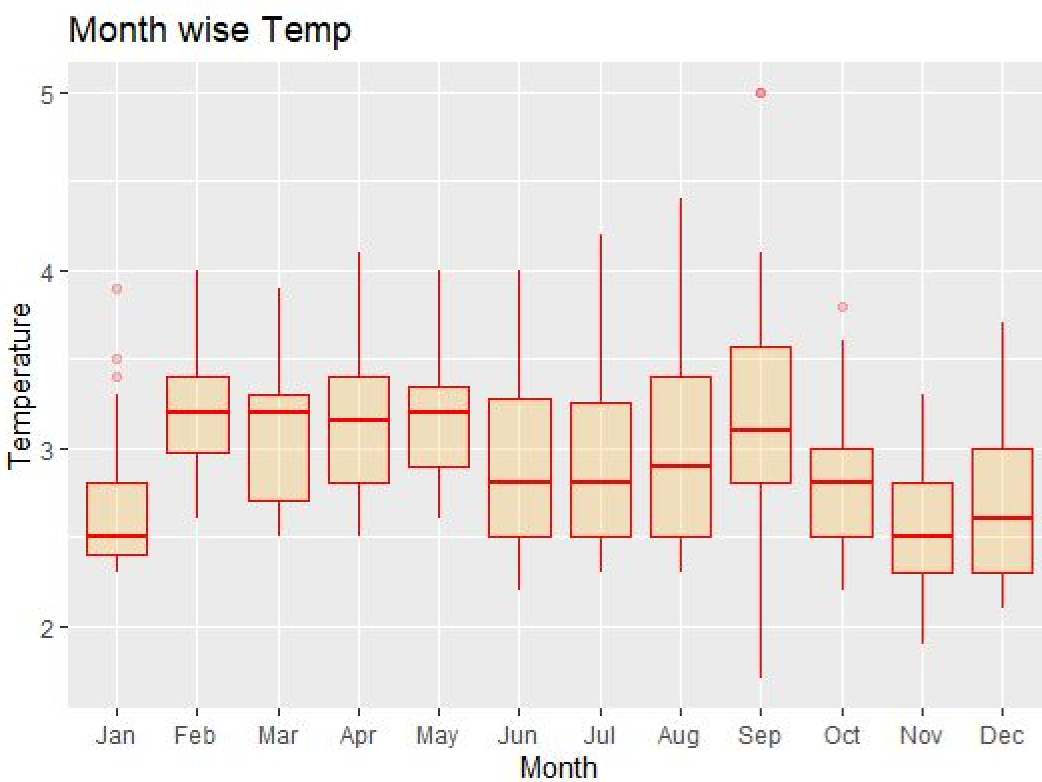
6.5 BoxPlot of Season wise Temperature



6.6 Monthly Avg Temperature



6.7 Boxplot of Monthly Temp



6.8 Month wise Temperature Plot