Cold Storage Notebook

Code

Setting the working directory

Hide

```
setwd("D:\\chandrima\\BACP - GreatLearning\\Fndmntls Of Business Stat - Project")

getwd()
```

```
[1] "D:/chandrima/BACP - GreatLearning/Fndmntls Of Business Stat - Project"
```

Problem Statement -1

Importing the raw data within R and saving it as part of a dataframe for analysis

Hide

```
coldstorage_df=read.csv("Cold_Storage_Temp_Data.CSV", header = TRUE)

dim(coldstorage_df)
```

```
[1] 365   4
```

Hide

```
##### The imported dataset has 365 rows and 4 columns #####
```

A brief look at the data

Hide

```
head(coldstorage_df,5)
```

	Season <fctr>	Month <fctr>	Date <int>	Temperature <dbl>
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

5 rows

Hide

```
NA
```

Hide

```
tail(coldstorage_df,5)
```

	Season <fctr>	Month <fctr>	Date <int>	Temperature <dbl>
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

5 rows

Code

Understanding the structure of the dataset

Hide

```
library(DataExplorer)

introduce(coldstorage_df)
```

r... <int>	colu... <int>	discrete_columns <int>	continuous_columns <int>	all_missing_columns <int>	total_miss
365	4	2	2	0	

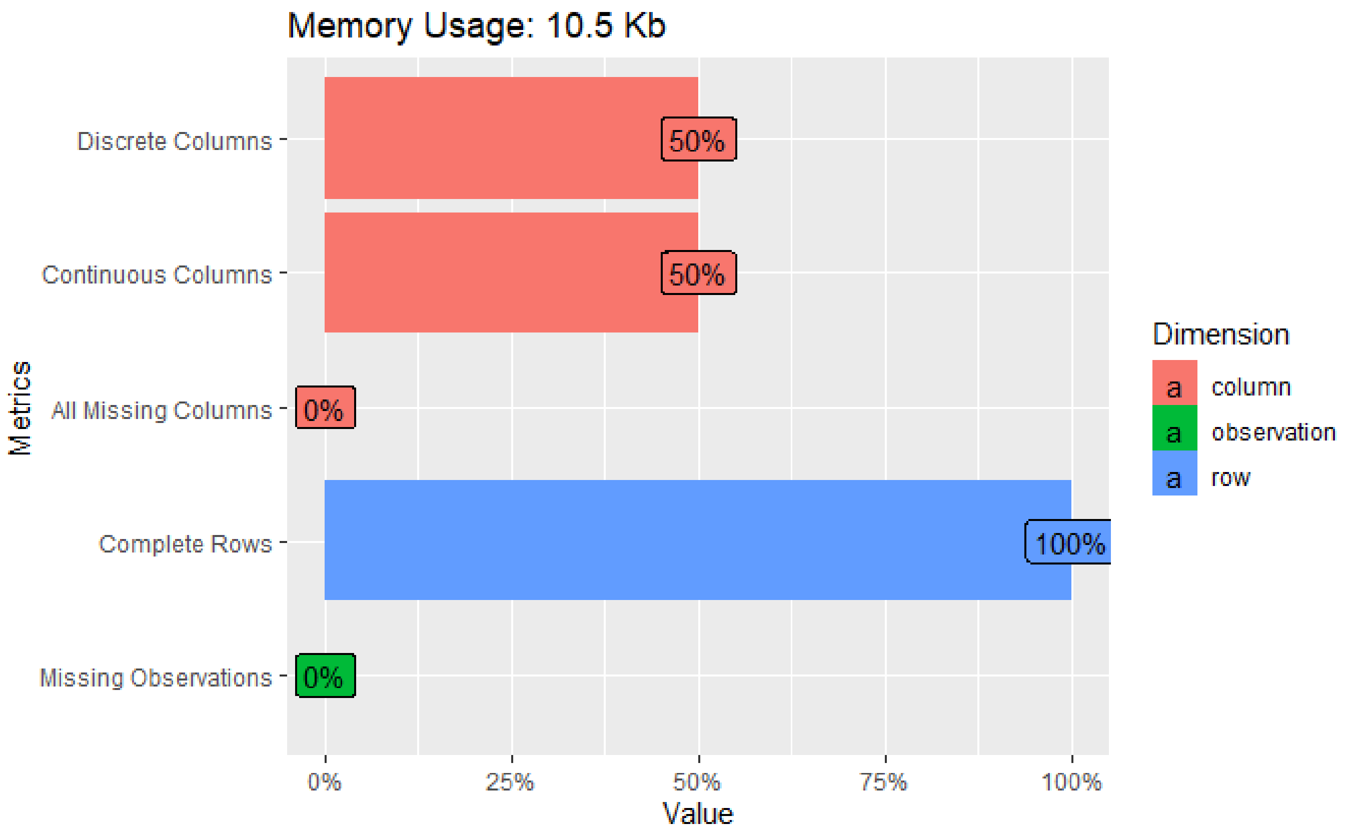
1 row | 1-6 of 9 columns

Hide

```
NA
```

Hide

```
plot_intro(coldstorage_df)
```



Checking the data types

Hide

```
str(coldstorage_df)
```

```
'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       : int  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 ...
```

Hide

```
##### Date has been treated as an integer variable which needs to be converted into date vari
able
##### For Month variable, the factor levels need to be re-ordered
```

Hide

```
summary(coldstorage_df)
```

```
   Season        Month          Date        Temperature
 Rainy :122   Aug    : 31   Min.   : 1.00   Min.   :1.700
 Summer:120   Dec    : 31   1st Qu.: 8.00   1st Qu.:2.500
 Winter:123   Jan    : 31   Median :16.00   Median :2.900
              Jul    : 31   Mean   :15.72   Mean   :2.963
              Mar    : 31   3rd Qu.:23.00   3rd Qu.:3.300
              May    : 31   Max.   :31.00   Max.   :5.000
              (Other):179
```

Changing the data type of Date variable

Hide

```
coldstorage_df$Date=as.Date(coldstorage_df$Date,origin = "2015-12-31")

str(coldstorage_df$Date)
```

```
 Date[1:365], format: "2016-01-01" "2016-01-02" "2016-01-03" "2016-01-04" "2016-01-05" "2016-
01-06" "2016-01-07" "2016-01-08" "2016-01-09" "2016-01-10" "2016-01-11" ...
```

Re-ordering the factor levels for the month variable

Hide

```
coldstorage_df$Month=ordered(coldstorage_df$Month, levels=c("Jan","Feb","Mar","Apr","May","Ju
n","Jul","Aug","Sep","Oct","Nov","Dec"))

str(coldstorage_df$Month)
```

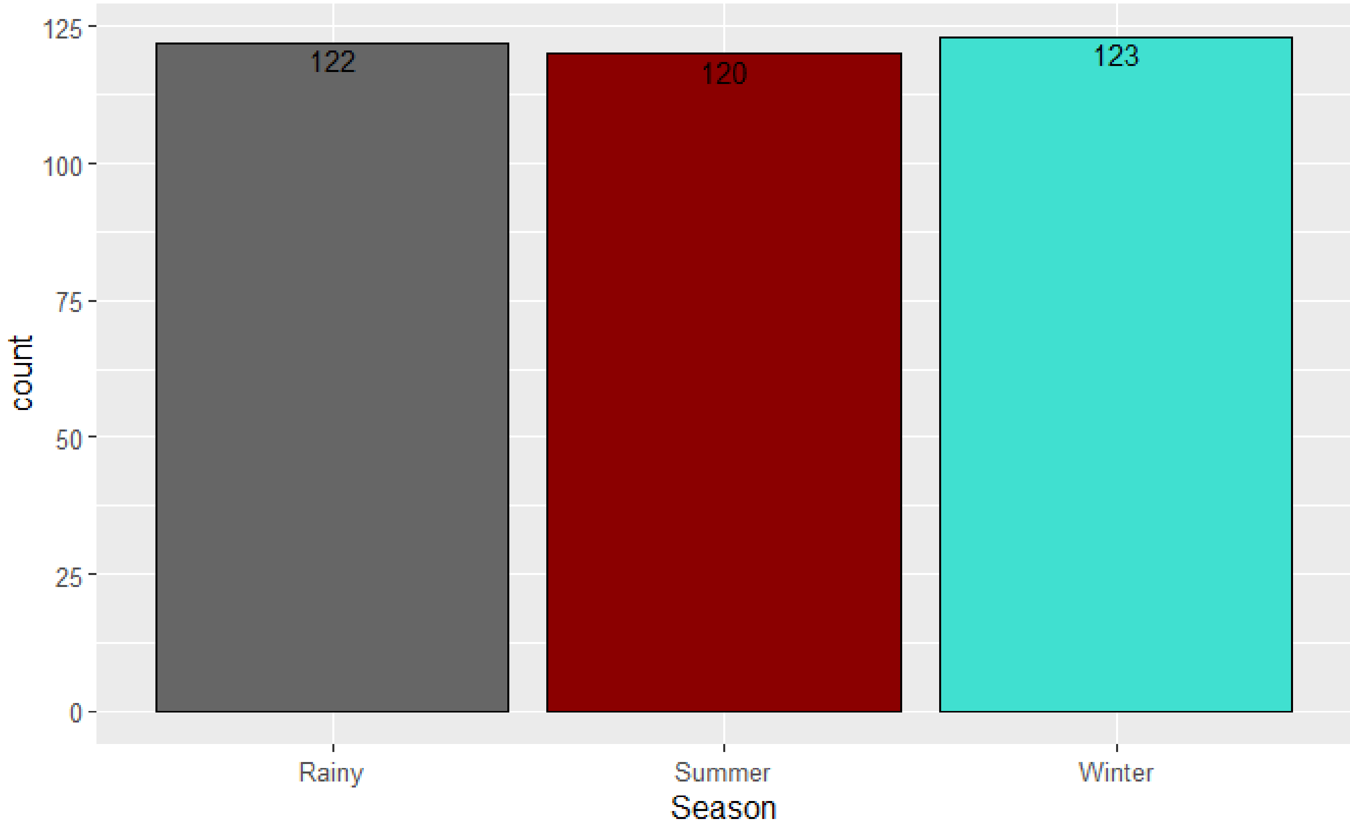
```
 Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<..: 1 1 1 1 1 1 1 1 1 1 ...
```

Graphical exploration of the dataset

Univariate Analysis of Season

Hide

```
library(ggplot2)

ggplot(coldstorage_df,aes(x=coldstorage_df$Season,fill=coldstorage_df$Season))+
     geom_bar(colour="black") + scale_fill_manual(values = c("Summer" = "darkred", "Rainy" =
"grey40", "Winter"="turquoise"))+
     xlab("Season")+ geom_text(stat='count', aes(label=..count..), vjust=1.2,colour="black")+
      theme(text = element_text(size = 12),legend.position="none")
```



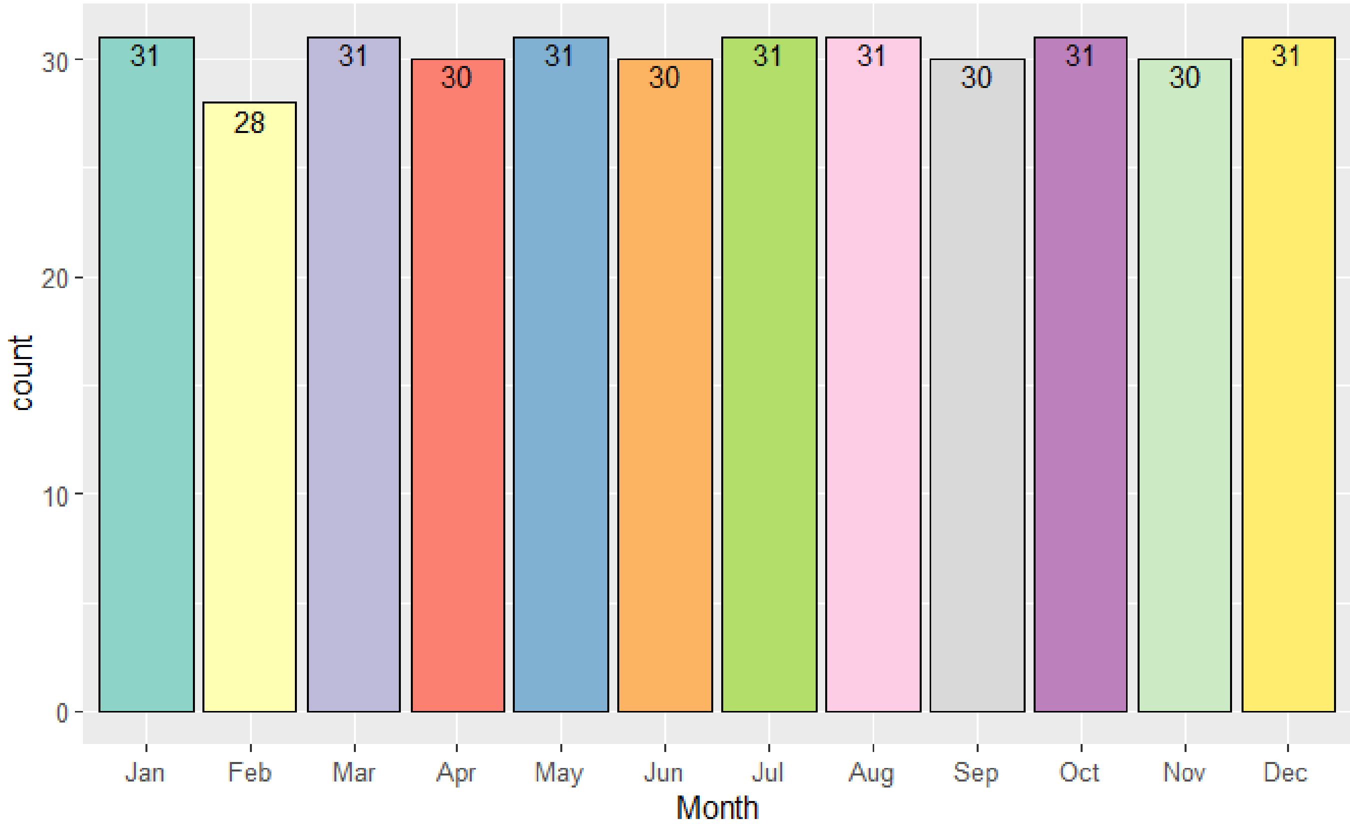
Hide

```
NA
NA
```

Univariate analysis of Month

Hide

```
ggplot(coldstorage_df,aes(x=coldstorage_df$Month,fill=coldstorage_df$Month))+
     geom_bar(colour="black") + scale_fill_brewer(palette = "Set3")+
     xlab("Month")+ geom_text(stat='count', aes(label=..count..), vjust=1.2,colour="black")+
     theme(text = element_text(size = 12),legend.position="none")
```



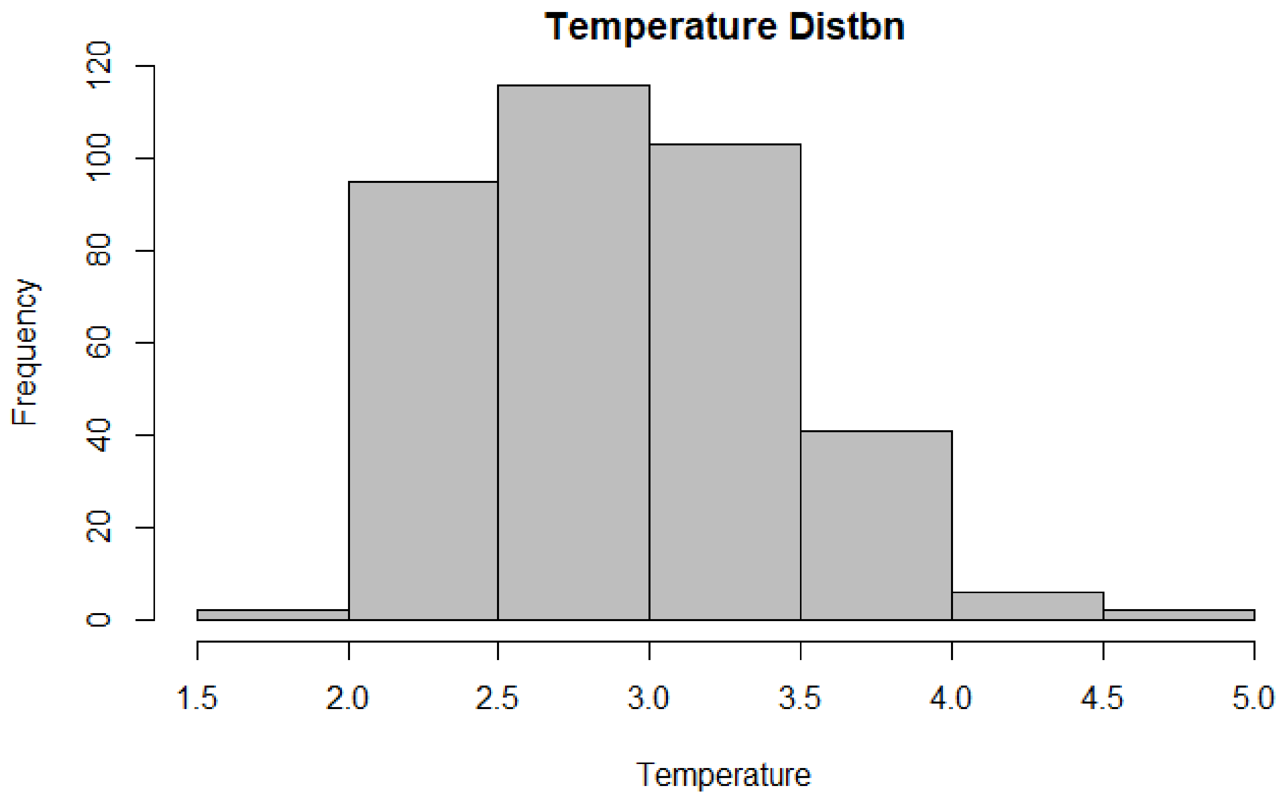
Hide

```
NA
NA
NA
```

Univariate analysis of Temperature

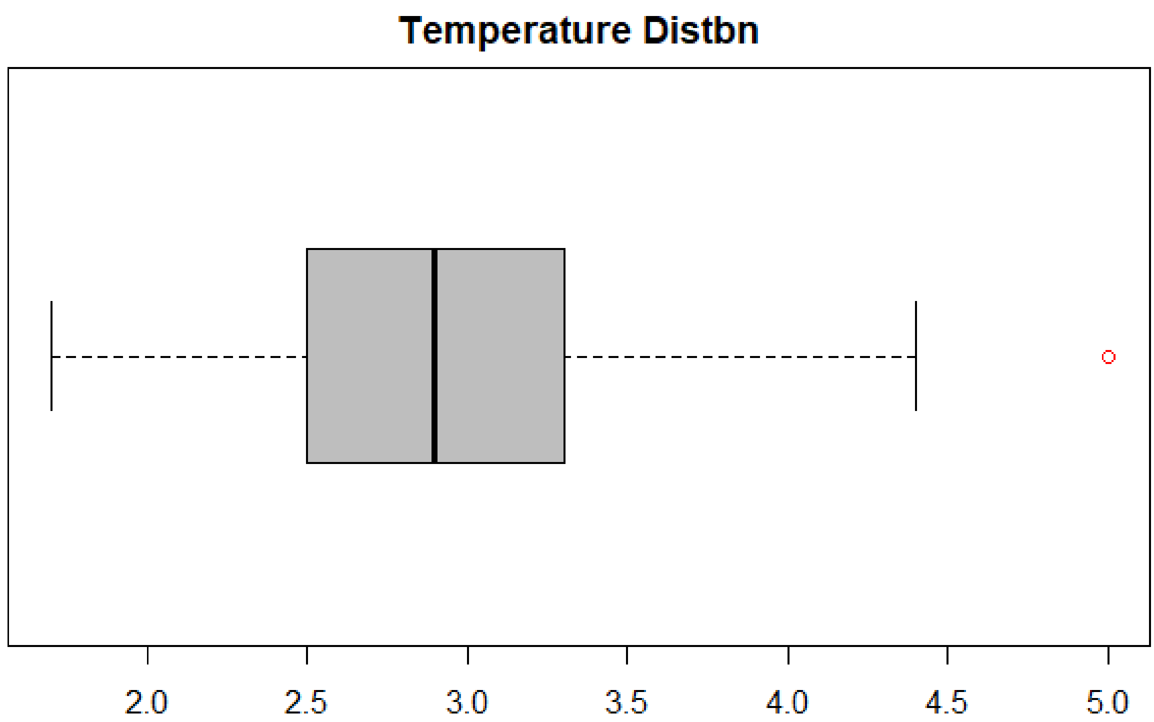
Hide

```
hist(coldstorage_df$Temperature, main = "Temperature Distbn", col = "grey", xlab = "Temperatu
re")
```



Hide

```
OutVals = boxplot(coldstorage_df$Temperature, main="Temperature Distbn",col = "grey",outcol =
"red" , horizontal = TRUE)$out
```



Hide

```
## Looks to be a normal distriution with outlier;
```

Hide

```
OutVals
```

```
[1] 5 5
```

Hide

```
##### 2 outlier present with the value of 5 #####
```

Exploring the data distribution for Temperature

Hide

```
library(psych)
describe(coldstorage_df$Temperature,IQR = T,quant = c(0.25,0.50,0.75))
```

vars <dbl>	n <dbl>	mean <dbl>	sd <dbl>	median <dbl>	trimmed <dbl>	mad <dbl>	min <dbl>	max <dbl>	range <dbl>
1	365	2.96	0.51	2.9	2.93	0.59	1.7	5	3.3

1 row | 1-10 of 17 columns

Hide

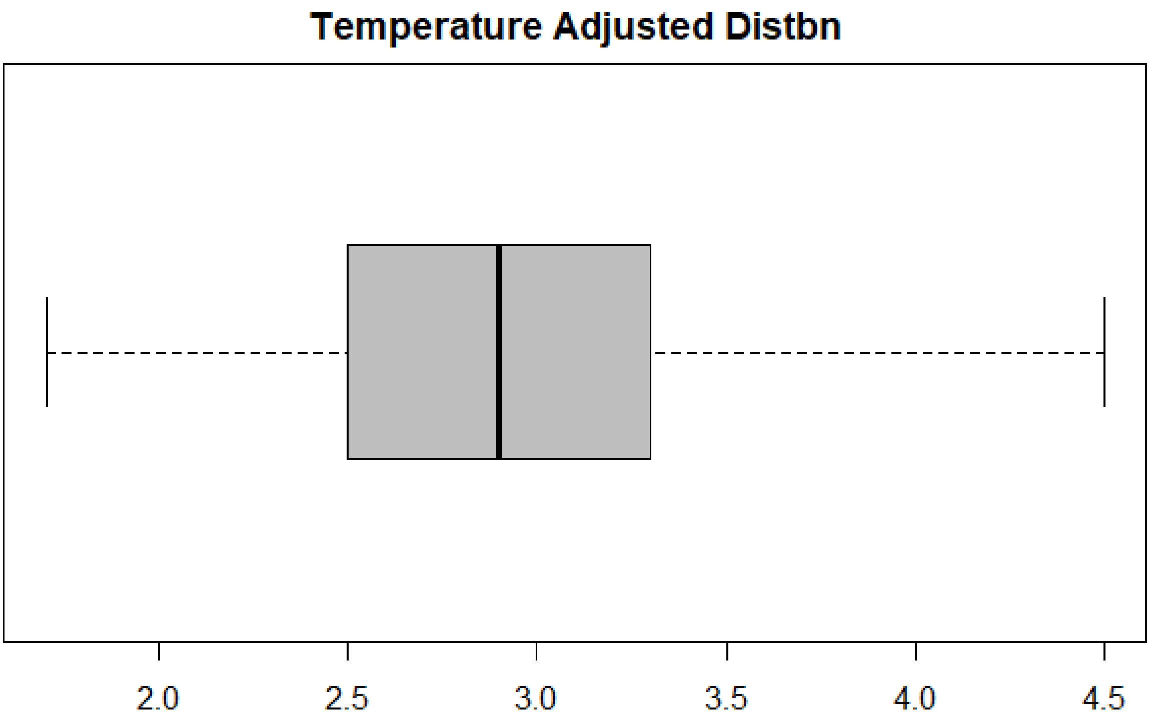
```
NA
```

Any value which lies beyond Q3+1.5IQR is an outlier. Instead of removing outliers altogether, the outlier values have been capped at Q3+1.5IQR level

A new field is introduced here - TemperatureAdjusted - where only the outlier values have been adjusted, retaining all other values of original Temperature

Hide

```
coldstorage_df$TemperatureAdjusted=ifelse(coldstorage_df$Temperature>4.5,4.5,coldstorage_df$T
emperature)

boxplot(coldstorage_df$TemperatureAdjusted, main="Temperature Adjusted Distbn",col = "grey",
 outcol ="red" , horizontal = TRUE)
```



Exploring the data distribution of the new field - TemperatureAdjusted

Hide

```
describe(coldstorage_df$TemperatureAdjusted,IQR = T,quant = c(0.25,0.50,0.75))
```

vars <dbl>	n <dbl>	mean <dbl>	sd <dbl>	median <dbl>	trimmed <dbl>	mad <dbl>	min <dbl>	max <dbl>	range <dbl>
1	365	2.96	0.5	2.9	2.93	0.59	1.7	4.5	2.8

1 row | 1-10 of 17 columns

Hide

```
NA
```

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

Hide

```
library(dplyr)
```

```
Attaching package: 惏牱dplyr惏柞

The following objects are masked from 惏牱package:stats惏柞:

    filter, lag

The following objects are masked from 惏牱package:base惏柞:

    intersect, setdiff, setequal, union
```

Hide

```
coldstorage_df %>% group_by(Season) %>% summarise(MeanBySeason=mean(Temperature),
                                                    MeanBySeason_Adjusted=mean(TemperatureAdjus
ted))
```

Season <fctr>	MeanBySeason <dbl>	MeanBySeason_Adjusted <dbl>
Rainy	3.039344	3.031148
Summer	3.153333	3.153333
Winter	2.700813	2.700813

3 rows

Hide

```
NA
```

Mean for the full year - taking the original Temperature values #####

Hide

```
mean(coldstorage_df$Temperature)
```

```
[1] 2.96274
```

Mean for the full year - taking the outlier adjusted Temperature values

Hide

```
mean(coldstorage_df$TemperatureAdjusted)
```

```
[1] 2.96
```

Standard deviation for the full year - taking the original Temperature values

Hide

```
sd(coldstorage_df$Temperature)
```

```
[1] 0.508589
```

Standard deviation for the full year - taking the outlier adjusted Temperature values

Hide

```
sd(coldstorage_df$TemperatureAdjusted)
```

```
[1] 0.4988338
```

Assume Normal distribution, what is the probability of temperature having fallen below 2 C?

Hide

```
pnorm(2, mean = 2.96274, sd=0.508589, lower.tail=TRUE)
```

```
[1] 0.02918142
```

Hide

```
## 2.92% probability that the temperature will fall below 2C
```

Assume Normal distribution, what is the probability of temperature having gone above 4 C?

Hide

```
pnorm(4, mean = 2.96274, sd=0.508589, lower.tail=FALSE)
```

```
[1] 0.02070079
```

Hide

```
## 2.07% probability that the temperature will go above 4c
```

Therefore it is statistically proven that the probability of the temperature falling below 2C or going above 4C is 0.02918142 + 0.02070079 = 0.04988221 or 4.99%. Therefore the penalty should be 10% of AMC

Problem Statement - 2

Importing the raw data within R and saving it as part of a dataframe for analysis

Hide

```
Mar2018_df=read.csv("Cold_Storage_Mar2018.CSV", header = TRUE)
```

Postulating the null and alternative hypothesis

Ho : mean <=3.9 Ha : mean >3.9

A brief look at the data

Hide

```
head(Mar2018_df,3)
```

	Season <fctr>	Month <fctr>	Date <int>	Temperature <dbl>
1	Summer	Feb	11	4.0
2	Summer	Feb	12	3.9
3	Summer	Feb	13	3.9

3 rows

Hide

```
NA
```

Understanding the structure of the dataset

Hide

```
introduce(Mar2018_df)
```

r… <int>	colu… <int>	discrete_columns <int>	continuous_columns <int>	all_missing_columns <int>	total_miss
35	4	2	2	0	

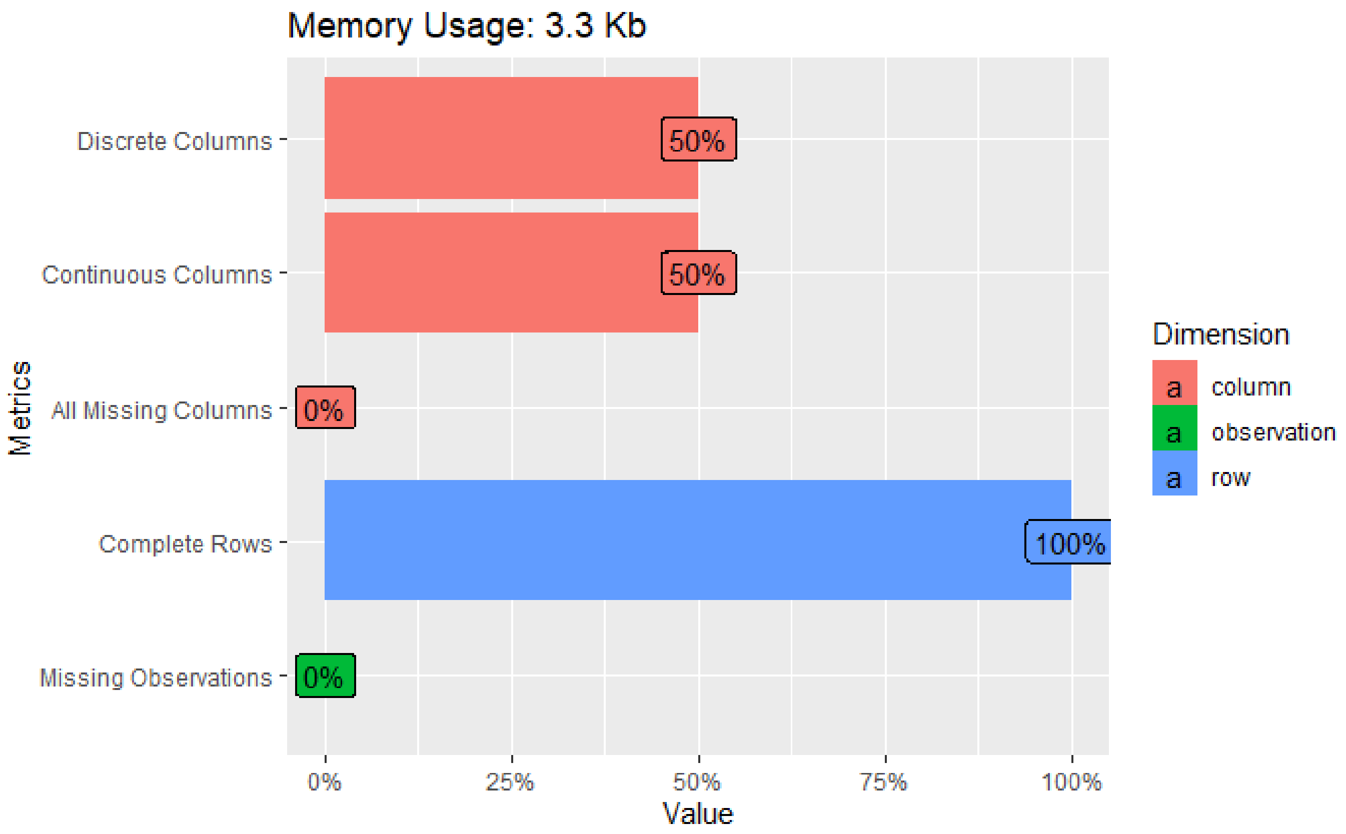
1 row | 1-6 of 9 columns

Hide

```
NA
```

Hide

```
plot_intro(Mar2018_df)
```



Checking the data types

Hide

```
str(Mar2018_df)
```

```
'data.frame':	35 obs. of  4 variables:
 $ Season     : Factor w/ 1 level "Summer": 1 1 1 1 1 1 1 1 1 1 ...
 $ Month      : Factor w/ 2 levels "Feb","Mar": 1 1 1 1 1 1 1 1 1 1 ...
 $ Date       : int  11 12 13 14 15 16 17 18 19 20 ...
 $ Temperature: num  4 3.9 3.9 4 3.8 4 4.1 4 3.8 3.9 ...
```

Hide

```
summary(Mar2018_df)
```

```
    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  
```

Changing the data type of Date variable

Hide

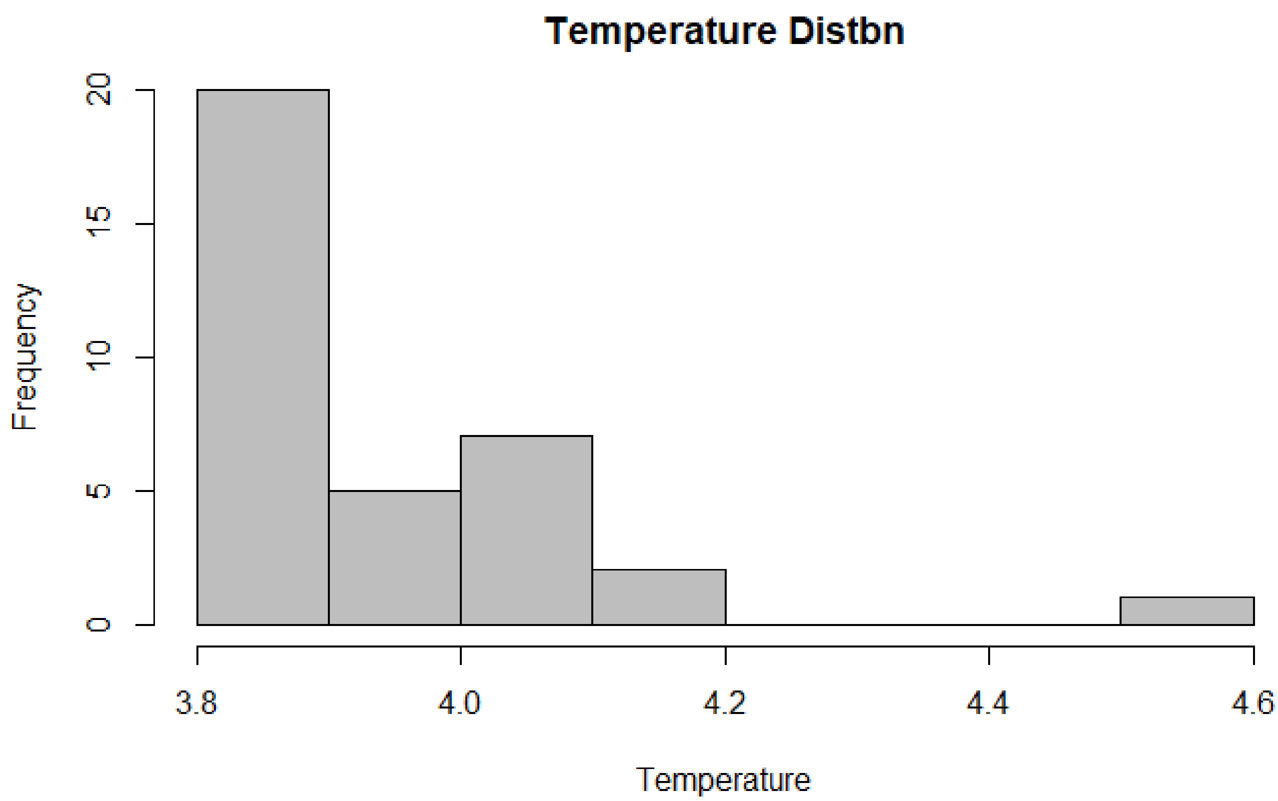
```
Mar2018_df$Date=as.Date(Mar2018_df$Date,origin = "2018-01-31")

class(Mar2018_df$Date)
```

```
[1] "Date"
```

Univariate analysis of temperature

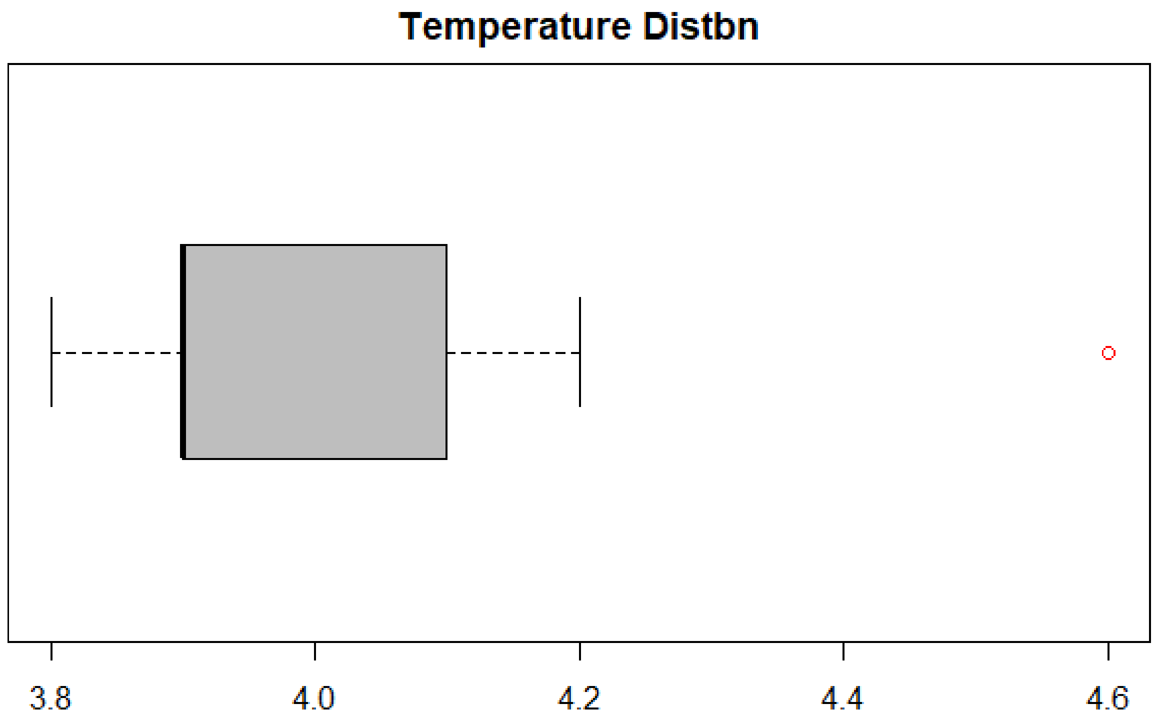
Hide

```
hist(Mar2018_df$Temperature, main = "Temperature Distbn",col = "grey", xlab = "Temperature")
```



Hide

```
outvals2=boxplot(Mar2018_df$Temperature, main="Temperature Distbn",col = "grey",outcol ="red"
, horizontal = TRUE)$out
```



Hide

```
##### Not a normal distribution and there is outlier present #####
```

Hide

```
outvals2
```

```
[1] 4.6
```

Hide

```
##### One outlier present with the value 4.6 #####
```

Exploring the data distribution for Temperature

Hide

```
describe(Mar2018_df$Temperature,IQR = T,quant = c(0.25,0.50,0.75,0.95,0.99,1))
```

vars <dbl>	n <dbl>	mean <dbl>	sd <dbl>	median <dbl>	trimmed <dbl>	mad <dbl>	min <dbl>	max <dbl>	range <dbl>
1	35	3.97	0.16	3.9	3.96	0.15	3.8	4.6	0.8

1 row | 1-10 of 20 columns

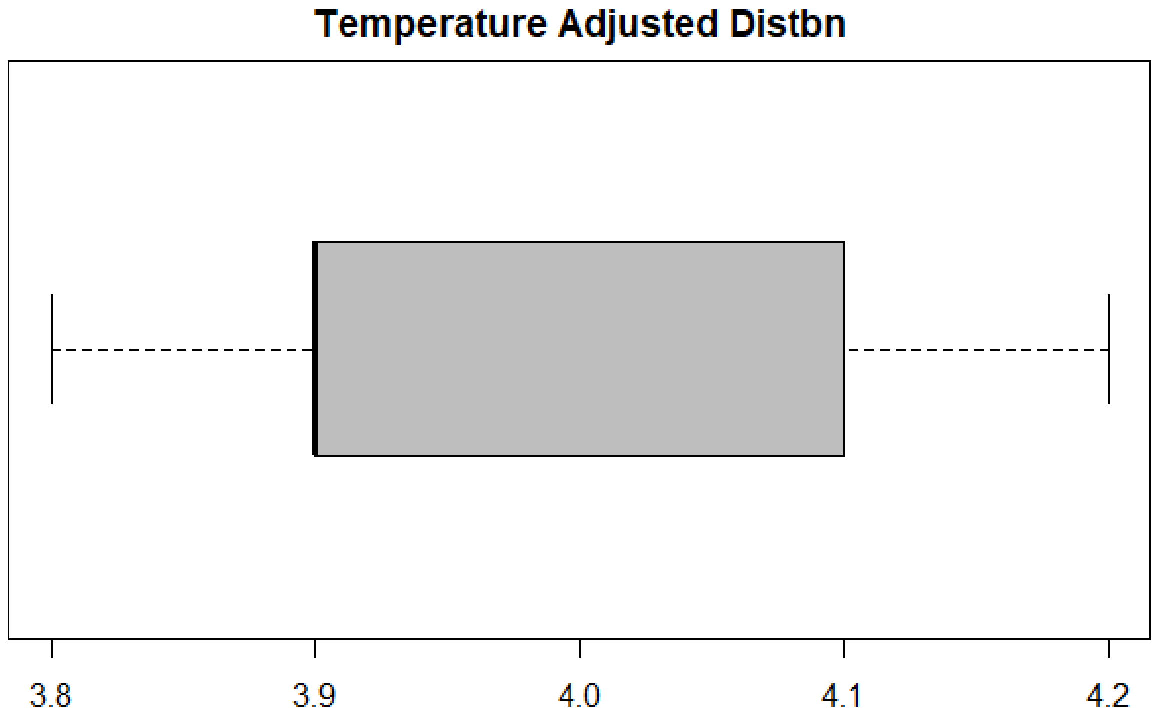
Hide

```
NA
```

Any value which lies beyond Q3+1.5IQR is an outlier. Instead of removing the outlier, the outlier value has been capped at 95th quantile level A new field is introduced here - TemperatureAdjusted - where only the outlier values have been adjusted, retaining all other values of original Temperature

Hide

```
Mar2018_df$TemperatureAdjusted=ifelse(Mar2018_df$Temperature>4.2,4.2,Mar2018_df$Temperature)

boxplot(Mar2018_df$TemperatureAdjusted, main="Temperature Adjusted Distbn",col = "grey",outco
l ="red" , horizontal = TRUE)
```



Finding the mean of the sample using both Temperature and TemperatureAdjusted

Hide

```
mean(Mar2018_df$Temperature)
```

```
[1] 3.974286
```

Hide

```
mean(Mar2018_df$TemperatureAdjusted)
```

```
[1] 3.962857
```

Finding the standard deviation of the sample using both Temperature and TemperatureAdjusted

Hide

```
sd(Mar2018_df$Temperature)
```

```
[1] 0.159674
```

Hide

```
sd(Mar2018_df$TemperatureAdjusted)
```

```
[1] 0.1238731
```

Using t-test for hypothesis testing using both Temperature and TemperatureAdjusted

Hide

```
t.test(Mar2018_df$Temperature,mu=3.9, alternative = "greater")
```

```

    One Sample t-test

data:  Mar2018_df$Temperature
t = 2.7524, df = 34, p-value = 0.004711
alternative hypothesis: true mean is greater than 3.9
95 percent confidence interval:
 3.928648      Inf
sample estimates:
mean of x 
 3.974286 
```

Hide

```
t.test(Mar2018_df$TemperatureAdjusted,mu=3.9, alternative = "greater")
```

```

    One Sample t-test

data:  Mar2018_df$TemperatureAdjusted
t = 3.002, df = 34, p-value = 0.0025
alternative hypothesis: true mean is greater than 3.9
95 percent confidence interval:
 3.927452      Inf
sample estimates:
mean of x 
 3.962857 
```

In both the cases since the p values of 0.004711 and 0.0025 are less than alpha=0.1, therefore the alternative hypothesis is accepted that the mean temperature in the Cold Storage Plant is exceeding the maximum accepted level of 3.9C