COLD STORAGE CASE STUDY

Table of Contents

Content	Page No
1.Project Objective	4
2.Assumptions	4
3.Data Analysis	4
3.1 Setting up the environment	5
3.1.1 Installing the necessary packages and invoking the corresponding libraries	5
3.1.2 Setting up the working directory	5
3.1.2.1.Getting information about current working directory	5
3.2 Importing the datasets	5
3.2.1 Importing the Cold_Storage_Temp_data.csv dataset	5
3.2.2 Importing the Cold_storage_mar2018.csv dataset	5
3.3 Finding the presence of missing (NA) values in the datasets(Both datasets)	5
3.4 Variable Identification	6
3.5 Variable Transformation	6
3.6 Finding outliers in the loaded dataset	8
3.6.1 Finding outlier using graphical way by using boxplot	8
3.6.2 Finding outlier using Rosner test	9
3.7 Calculating mean cold storage Temperature for Summer, Winter & Rainy season	10
3.8 Calculating mean for full year	10
3.9 Finding Standard deviation for the full year	10
3.10 Calculating probability	11
3.10.1 probability of temperature falling below 2 degree Celsius	11
3.10.2 probability of temperature rising above 4	11

degree Celsius	
3.11 Stating the hypothesis and performing calculation using z test and t test	11
3.11.1 Hypothesis formulation	11
3.11.2 Calculating t test	12
3.11.3 Calculating z test	13
3.12 Drawing inferences based on both the tests(z test and t test)	14
3.13 Source Code	15

1.Project Objective

The prime objective of the project is to explore two provided datasets- " **Cold_Storage_Temp_data.csv "** and " **Cold_storage_mar2018.csv "** in R and generate insights about the datasets.

The report consists of following contents:

- Importing the datasets in R
- Understanding structure of the datasets
- Performing basic checks on the datasets
- Graphical projections
- Computing probabilities
- Drawing insights from the datasets

2.Assumptions

We assume that the **Cold_Storage_Temp_data.csv dataset** and **Cold_Storage_Mar2018.csv** follows a normal distribution(Bell curve distribution)

3.Data Analysis:

Data analysis activity consist of following steps:

1. Setting up environment
2. Importing datasets
3. Finding missing(NA) values in the datasets
4. Variable Identification
5. Variable Transformation.
6. Outlier detection (Using both graphical and numerical way)
7. Calculating mean cold storage temperature for Summer, Winter and Rainy Season
8. Calculating mean for full year
9. Finding standard deviation for full year
10. Calculating probability and stating the conclusion regarding the penalty
11. Stating the hypothesis and performing calculation using z test and t test

12. Drawing Inference based on both the test.
13. Source Code

3.1 Setting up the environment

3.1.1 Installing the necessary packages and invoking the corresponding libraries

- readr : library(readr)
- ggplot2 : library(ggplot2)
- outliers : library(ouliers)
- EnvStats : library(EnvStats)

3.1.2 Setting up the working directory

3.1.2.1.Gettiing information about current working directory

Query: getwd()

3.1.2.2.Setting up the working directory

Query: setwd("D:/RGreatlearning")

3.2 Importing the datasets

3.2.1 Importing the Cold_Storage_Temp_data.csv dataset

Query : ColdStorage = read_csv("K2_Cold_Storage_Temp_Data (1).csv")

3.2.2 Importing the Cold_storage_mar2018.csv dataset

Query: COldStorage_2018 = read_csv("K2_Cold_Storage_Mar2018.csv")

3.3 Finding the presence of missing (NA) values in the datasets(Both datasets)

Query : anyNA(ColdStorage)

```
[1] FALSE
```

Query: anyNA(COldStorage_2018)

```
[1] FALSE
```

From this we can conclude that both datasets doesn't have missing(NA) values present

3.4 Variable Identification

For this project we have used following R functions:

- **read_csv()**: This function is a part of readr package. The main purpose of this function is to load the dataset in the .CSV format in R
- **dim()**: used for calculating dimensions of the loaded dataset.
- **Str()**: checking structure of the dataset
- **Summary()**: getting summary of the datasets.
- **as.factor()**: for converting the selected column's datatype into factor.
- **boxplot()**: for plotting boxplot
- **rosnerTest()**: for performing rosner test for outlier detection
- **anyNA()**: checking for missing values
- **mean()**: computing mean
- **sd()**: computing standard deviation
- **pnorm()**: Computing probability in case of normal distribution

3.5 Variable Transformation

In the loaded dataset after checking the dataset using summary() function it appears that the columns Seasons and Months are of character type which doesn't provide any useful information.

For Cold_Storage_Temp_data.csv dataset

```
    Season                  Month                  Date
Temperature
 Length:365            Length:365           Min.   : 1.00
Min.   :1.700
 Class :character      Class :character     1st Qu.: 8.00
1st Qu.:2.500
 Mode  :character      Mode  :character     Median :16.00
Median :2.900
                                            Mean   :15.72
Mean   :2.963
                                            3rd Qu.:23.00
3rd Qu.:3.300
                                            Max.   :31.00
Max.   :5.000
```

For Cold_storage_mar2018.csv dataset

```
    Season                  Month                     Date
Temperature
 Length:35             Length:35            Min.   : 1.0   M
in.   :3.800
 Class :character      Class :character     1st Qu.: 9.5   1
st Qu.:3.900
 Mode  :character      Mode  :character     Median :14.0   M
edian :3.900
                                            Mean   :14.4   M
ean   :3.974
                                            3rd Qu.:19.5   3
rd Qu.:4.100
                                            Max.   :28.0   M
ax.   :4.600
```

As these two columns hold collection of values i.e in case of Seasons it hold values like Summer, Winter & Rainy while for Months it hold values like Jan,Feb,Mar etc. let's convert these two into factors

ColdStorage$Season = as.factor(ColdStorage$Season)

ColdStorage$Month = as.factor(ColdStorage$Month)

COldStorage_2018$Season = as.factor(COldStorage_2018$Season)

COldStorage_2018$Month = as.factor(COldStorage_2018$Month)

Now let's check summary of both datasets after appropriate transformation

For Cold_Storage_Temp_data.csv dataset

```
    Season          Month              Date            Temperatur
e
 Rainy :122    Aug    : 31    Min.   : 1.00    Min.   :1.7
00
 Summer:120    Dec    : 31    1st Qu.: 8.00    1st Qu.:2.5
00
 Winter:123    Jan    : 31    Median :16.00    Median :2.9
00
               Jul    : 31    Mean   :15.72    Mean   :2.9
63
```

```
                 Mar     : 31    3rd Qu.:23.00    3rd Qu.:3.3
00
                 May     : 31    Max.    :31.00    Max.    :5.0
00
                 (Other):179
```

For Cold_storage_mar2018.csv dataset

```
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
```

NOTE:- In this project after loading the dataset, we have attached the dataset using attach() function because of which we can directly used variable names in the dataset.

For example. Using "Temperature" directly rather than using "ColdStorage$Temperature " !.

3.6 Finding outliers in the loaded dataset

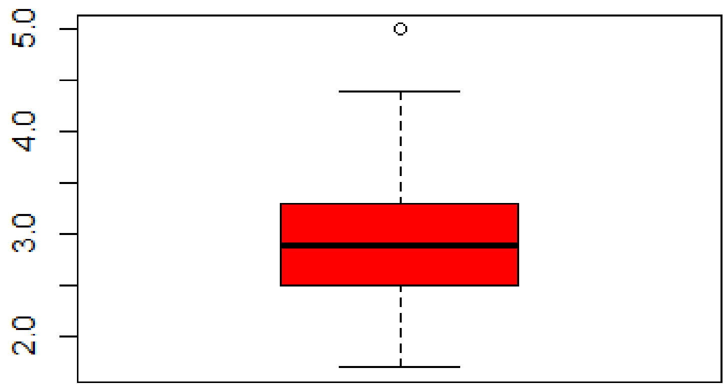
3.6.1 Finding outlier using graphical way by using boxplot

Technically , Outlier is a value which is far different from other values present in the dataset. Selection of outlier is a subjective decision however presence of outlier affects the mean .

Let's find the outlier by using boxplot

Boxplot:

Query: boxplot(Temperature)



The boxplot shows that there is indeed an outlier present in the dataset which is the value 5. Just to be sure lets find outlier by using rosner test

3.6.2 Finding outlier using Rosner test

Rosner test is an outlier detection test which is used for detecting outliers present in dataset. Rosner test can detect up to 10 outliers.

Query: rosnerTest(Temperature,k=3,warn = F)

```
Results of Outlier Test
-------------------------

Test Method:                     Rosner's Test for Outl
iers

Hypothesized Distribution:       Normal

Data:                            Temperature

Sample Size:                     365

Test Statistics:                 R.1 = 4.005710
                                 R.2 = 4.102898
                                 R.3 = 2.975142
```

```
Test Statistic Parameter:        k = 3

Alternative Hypothesis:          Up to 3 observations a
re not
                                 from the same Distribu
tion.

Type I Error:                    5%

Number of Outliers Detected:     2

  i   Mean.i      SD.i Value Obs.Num    R.i+1 lambda.i+
1 Outlier
1 0 2.962740 0.5085890   5.0     252 4.005710   3.77822
3    TRUE
2 1 2.957143 0.4979059   5.0     263 4.102898   3.77746
8    TRUE
3 2 2.951515 0.4868624   4.4     234 2.975142   3.77671
1   FALSE
```

The Rosner test for outlier detection indeed confirm that there are outliers present in the dataset(Cold_Storage_Temp_data.csv) which is value 5.But in this case we are going to ignore the outliers since maintaining a constant temperature within a particular range (in this case between 2 – 4 degree) is challenging task and sometimes temperature might fall above or below the recommended range. So it is possible that temperature may have cross the upper permissible limit (4 degree).

3.7 Calculating mean cold storage Temperature for Summer, Winter & Rainy season

In order to compute the mean cold storage temperature for 3 seasons we have used tapply() and aggregrate() function

3.7.1 Mean cold storage Temperature for Summer, Winter & Rainy season using tapply() function

Query: tapply(Temperature,Season,mean)

```
Rainy    Summer   Winter
3.039344 3.153333 2.700813
```

3.7.2 Mean cold storage Temperature for Summer, Winter & Rainy season using aggregate() function

Query: aggregate(Temperature~Season,ColdStorage,mean)

```
  Season Temperature
1  Rainy    3.039344
2 Summer    3.153333
3 Winter    2.700813
```

As we can see that the mean cold storage temperature for Rainy season is 3.039344 followed by 3.15333 , 2.700813 for Summer and Winter season

3.8 Calculating mean for full year

Query: mean(Temperature)

```
[1] 2.96274
```

By calculation it can be specified that mean cold storage temperature value for full year is 2.96274

3.9 Finding Standard deviation for the full year

Query: sd(Temperature)

```
[1] 0.508589
```

The standard deviation for the full year is found to be 0.508589

3.10 Calculating probability

Assuming that the distribution is a normal distribution,

3.10.1 Probability of temperature falling below 2 degree celcius

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

```
[1] 0.02918142
```

Here we have to compute probability for temperature below 2 degree celcius and hence for that we have set the lower.tail property to TRUE.Thus, The probability of temperature falling below 2 degree celsius is 0.02918142 ~ 2.91

3.10.2 Probability of temperature rising above 4 degree celcius

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

```
[1] 0.02070079
```

Here we have to compute the probability of temperature rising above 4 degree celsius and hence for that we have set the lower.tail property to FALSE. Thus the probability of temperature rising above 4 degree celsius is 0.0207079~ 2.07.

"As per the condition mentioned in the problem statement, penalty will be 10% of Annual Maintenance Case(AMC)"

3.11 Stating the hypothesis and performing calculation using z test and t test

3.11.1 Hypothesis formulation:

As mentioned in the problem statement " ***As a safety measure, the Supervisor has been vigilant to maintain the temperature below 3.9 deg C.***"

Based on this let's formulate Alternative Hypothesis(Ha) and Null Hypothesis (H0).

Alternative Hypothesis (Ha): "The Supervisor has been vigilant to maintain the temperature below 3.9 deg C"

Null Hypothesis (H0): "The Supervisor has been vigilant to maintain the temperature greater than or equal to 3.9 deg C"

3.11.2 Calculating t test

Variables used in computing t test:

- **xbar** = Estimated value
- **Mu** = Sample mean
- **S** = Standard deviation in terms of sample

- **n** = No of observation which are present in the sample.

Detailed Explaination:

- As mentioned in the problem statement that due to increase in the no .of complaints by end consumers ,as a safety measure the supervisor is vigilant to maintain temperature below 3.9 deg C which is an upper acceptable range. So in this case based on the null hypothesis that we have formulated, let's assume the "Mu" value as 3.9. Hence **Mu = 3.9**
- xbar is computed by applying mean() function to Temperature column in the K2_Cold_Storage_Mar2018.csv dataset which is computed as 3.974286. Hence **xbar = 3.974286**
- Standard deviation for this sample is computed by applying sd() function to Temperature column in the K2_Cold_Storage_Mar2018.csv dataset which is computed as 0.159674. Hence **S = 0.159674.**
- Since the dataset K2_Cold_Storage_Mar2018.csv contains 36 observations and therefore value of n should be 36. Hence **n = 36**

Formula for performing t test:

tstat = xbar – Mu / (S – (n^0.5))

Based on the above formula, by substituting the given value the value of tstat is found to be 2.791401

Computing P value:

Technically P-value which is also called as **probability value (also known as Actual Significance level)** is basically the actual risk or actual value of significance by which null hypothesis is rejected.

Pvalue = pt(tstat,35) where 35 is the degree of freedom a random variable can have

Based on the above computation the Pvalue is found to be 0.9957784

3.11.3 Calculating z test

Variables used in computing z test:

- **xbar** = Estimated value
- **Mu** = Sample mean
- **Sigma** = Standard deviation

- **n** = no. of observation

Detailed Explaination:

- As mentioned in the problem statement that due to increase in the no .of complaints by end consumer,as a safety measure the supervisor is vigilant to maintain temperature below 3.9 deg C which is an upper acceptable range. So in this case based on the null hypothesis that we have formulated, let's assume the "Mu" value as 3.9. Hence **Mu = 3.9**
- xbar is computed by applying mean() function to Temperature column in the K2_Cold_Storage_Mar2018.csv dataset which is computed as 3.974286. Hence **xbar = 3.974286**
- Standard deviation for this sample is computed by applying sd() function to Temperature column in the K2_Cold_Storage_Mar2018.csv dataset which is computed as 0.159674. Hence **S = 0.159674**
- Since the dataset K2_Cold_Storage_Mar2018.csv contains 36 observations and therefore value of n should be 36. Hence **n = 36**

Formula for performing z test:

zstat = xbar – Mu / (S – (n^0.5))

Based on the above formula, by substituting the given value the value of zstat is found to be 2.791401

Computing P value:

Technically P-value which is also called as **probability value (also known as Actual Significance level)** is basically the actual risk or actual value of significance by which null hypothesis is rejected.

Pvalue = pt(zstat,35) where 35 is the degree of freedom a random variable can have

Based on the above computation the Pvalue is found to be 0.9957784

3.12 Drawing inferences based on both the tests(z test and t test)

Based on the computation of both z test and t test, following inferences can be drawn:

- The P-value for both the z test and t test is found to be 0.9957784. As a rule, if the value of P-value is greater than the Alpha then in such conditons, Null Hypothesis is accepted. So in this case our formulated Null hypothesis is accepted since the P-value is greater than alpha(0.1)
- z test and t test both works hand-in-hand with each other however the point at which both these differ is their practical implementation. Technically z test is computed for population data but it can also be used for sample data provided that the value of n should be greater than 30.(n > 30). While t test is more suitable for the sample size when no of observations is less than 30
- For large sized samples, the t test gives similar computation results as that of z test.Hence in this condition we are getting same values for z test and t test

3.13 Source Code

```
###Data Analysis-Cold Storage Case Study#########

#Installing the necessary packages

install.packages("readr")

install.packages("ggplot2")

install.packages("outliers")

install.packages("EnvStats")

##using the installed libraries

library(readr)
```

```
> library(readr)
```

```
library(ggplot2)
```

```
> library(ggplot2)
Registered S3 methods overwritten by 'ggplot2':
  method         from
  [.quosures     rlang
  c.quosures     rlang
  print.quosures rlang
```

```
library(outliers)
```

```
> library(outliers)
```

```
library(EnvStats)
```

```
> library(EnvStats)

Attaching package: ‘EnvStats’

The following objects are masked from ‘package:stats’:

    predict, predict.lm

The following object is masked from ‘package:base’:

    print.default
```

```
##Getting information about current working directory
getwd()
```

```
> getwd()
[1] "D:/RGreatlearning"
```

```
##Setting the working directory
setwd("D:/RGreatLearning")
```

```
> setwd("D:/RGreatLearning")
```

```
##loading the dataset into R
ColdStorage = read_csv("K2_Cold_Storage_Temp_Data (1).csv")
```

```
> ColdStorage = read_csv("K2_Cold_Storage_Temp_Data (1)
.csv")
Parsed with column specification:
cols(
  Season = col_character(),
  Month = col_character(),
```

```
  Date = col_double(),
  Temperature = col_double()
)
```

COldStorage_2018 = read_csv("K2_Cold_Storage_Mar2018.csv")

```
> COldStorage_2018 = read_csv("K2_Cold_Storage_Mar2018.
csv")
Parsed with column specification:
cols(
  Season = col_character(),
  Month = col_character(),
  Date = col_double(),
  Temperature = col_double()
)
```

##Checking the structure of the loaded dataset

str(ColdStorage)

```
> str(ColdStorage)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame'
:	365 obs. of  4 variables:
 $ Season     : chr  "Winter" "Winter" "Winter" "Winter
" ...
 $ Month      : chr  "Jan" "Jan" "Jan" "Jan" ...
 $ Date       : num  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 ...
 - attr(*, "spec")=
  .. cols(
  ..   Season = col_character(),
  ..   Month = col_character(),
  ..   Date = col_double(),
  ..   Temperature = col_double()
  .. )
```

str(COldStorage_2018)

```
> str(COldStorage_2018)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame'
:	35 obs. of  4 variables:
```

```
 $ Season     : chr  "Summer" "Summer" "Summer" "Summer
"  ...
 $ Month      : chr  "Feb" "Feb" "Feb" "Feb" ...
 $ Date       : num  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 ..
.
 - attr(*, "spec")=
  .. cols(
  ..   Season = col_character(),
  ..   Month = col_character(),
  ..   Date = col_double(),
  ..   Temperature = col_double()
  .. )
```

```
##Checking the summary of the loaded dataset

summary(ColdStorage)
```

```
> summary(ColdStorage)
    Season             Month                Date        Temperature  
 Length:365         Length:365         Min.   : 1.00   Min.   :1.700  
 Class :character   Class :character   1st Qu.: 8.00   1st Qu.:2.500  
 Mode  :character   Mode  :character   Median :16.00   Median :2.900  
                                       Mean   :15.72   Mean   :2.963  
                                       3rd Qu.:23.00   3rd Qu.:3.300  
                                       Max.   :31.00   Max.   :5.000  
>
```

```
summary(COldStorage_2018)
```

```
> summary(COldStorage_2018)
    Season             Month                Date     
Temperature
 Length:35          Length:35          Min.   : 1.0   M
in.   :3.800  
 Class :character   Class :character   1st Qu.: 9.5   1
st Qu.:3.900  
 Mode  :character   Mode  :character   Median :14.0   M
edian :3.900  
                                       Mean   :14.4   M
ean   :3.974  
                                       3rd Qu.:19.5   3
rd Qu.:4.100  
```

```
                                        Max.   :28.0   M
ax.   :4.600
```

##Converting the datatype of season and Month into Factors

ColdStorage$Season = as.factor(ColdStorage$Season)

ColdStorage$Month = as.factor(ColdStorage$Month)

COldStorage_2018$Season = as.factor(COldStorage_2018$Season)

COldStorage_2018$Month = as.factor(COldStorage_2018$Month)

##Viewing summary after changing datatype

```
> summary(ColdStorage)
    Season          Month             Date           Temperatur
e
 Rainy :122   Aug     : 31   Min.   : 1.00   Min.   :1.7
00
 Summer:120   Dec     : 31   1st Qu.: 8.00   1st Qu.:2.5
00
 Winter:123   Jan     : 31   Median :16.00   Median :2.9
00
              Jul     : 31   Mean   :15.72   Mean   :2.9
63
              Mar     : 31   3rd Qu.:23.00   3rd Qu.:3.3
00
              May     : 31   Max.   :31.00   Max.   :5.0
00
              (Other):179
```

####Computing Mean cold storage temperature for summer, winter and rainy season

aggregate(Temperature~Season,ColdStorage,mean)

```
  Season Temperature
1  Rainy    3.039344
2 Summer    3.153333
3 Winter    2.700813
```

```
##Computing the same using tapply function

tapply(Temperature,Season,mean)
```

```
Rainy    Summer   Winter
3.039344 3.153333 2.700813
```

```
##Calculating mean for full year

mean(Temperature)
```

```
> mean(Temperature)
[1] 3.974286
```

```
##Calculating standard deviation for Full year

sd(Temperature)
```

```
> sd(Temperature)
[1] 0.159674
```

```
##calculating t test


xbar = mean(Temperature)

xbar

Mu = 3.9

S =sd(Temperature)

S

n = 36

tstat = (xbar - Mu)/ (S/(n^0.5))

tstat

Pvalue = pt(tstat,35)

Pvalue
```

```
> xbar = mean(Temperature)
> xbar
```

```
[1] 3.974286
> Mu = 3.9
> S =sd(Temperature)
> S
[1] 0.159674
> n = 36
> tstat = (xbar - Mu)/ (S/(n^0.5))
> tstat
[1] 2.791401
> Pvalue = pt(tstat,35)
> Pvalue
[1] 0.9957784
```

```
##Since p value is greater than alpha null hypothesis is accepted

##Calculating Z test
xbar = mean(Temperature)
xbar
Mu = 3.9
Sigma = sd(Temperature)
Sigma
n = 36
zstat = (xbar1 - Mu)/ (Sigma / (n^0.5))
zstat
Pvalue = pt(zstat,35)
Pvalue
```

```
> ##Calculating Z test
> xbar = mean(Temperature)
> xbar
[1] 3.974286
> Sigma = sd(Temperature)
> Sigma
[1] 0.159674
> n = 36
```

```
> zstat = (xbar - Mu)/ (Sigma / (n^0.5))
> zstat
[1] 2.791401
> Pvalue = pt(zstat,35)
> Pvalue
[1] 0.9957784
```

Since p value is greater than alpha Null hypothesis is accepted.