MATH2349 Semester 1, 2019

Assignment 3

Brian Steven Rathod (s3760875)

Required packages

```
library(readr)
library(EnvStats)
library(dplyr)
library(tidyr)
library(MVN)
```

Executive Summary

For the purpose of this assignment, two data sets were merged to meet the requirements. As part of the steps of preprocessing, a number of functions and packages were used including readr, envstats, mvn, tidyr and dplyr. Some of the essential functions included if_else() and mutate() which were utilised to create new variables. A mutated variable summing two other numeric variables was added to the dataset. Another categorical variable named Location was added, which sorted observations based on the corresponding values in Area_Name column. Redundant variables were dropped and the data set was re-ordered.

Head(), summary() and str() functions were used to summarize the structure of variables and inspect the attributes. Missing values and inconsistencies were checked column-wise. Due to the large number of NA values, they were plotted as 0 to facilitate functions and prevent errors.

A large number of outliers were detected in all numeric variables, using boxplot function. This occurence was because of a large number of 0 vaues per variable as well as a low mean and IQR. Further inspection revealed, that this was due to the specific nature of data at hand rather than any error in data collection or sampling. Imputing wasn’t performed on the outliers, since it would have led to distortion of information.

Data transformation was applied on the variable “CD_Hospitalisation_Treatment”, since it had a steep skew to the right. Natural logarithm was used to transform the variable.

Data

The 2 data sets chosen for this assignment were related to crime in India. Data set “hosp” contains information about deaths of criminals during Hospitalisation/Treatment, while the data set “prod” contains information about deaths in Custody during production/process in courts/journey connected with investigation. The data was sourced from Kaggle user Rajanand(https://www.kaggle.com/rajanand/crime-in-india/ (https://www.kaggle.com/rajanand/crime-in-india/))

Hosp has 5 variables - Area_Name, Year, Group_Name, Sub_Group_Name and CD_Hospitalisation_Treatment.

Prod has 11 variables including Area_Name, Year, Group_Name, Sub_Group_Name, CD_Deaths_Reported. It also contains information about cases conducted, cases registered, judicial enquiries, magisterial enquiries, policemen charged and convicted.

```
prod<- read_csv("D:/Sem 1/PPC/A 3/40_03_Custodial_death_during_production.csv")
```

```
## Parsed with column specification:
## cols(
##   Area_Name = col_character(),
##   Year = col_double(),
##   Group_Name = col_character(),
##   Sub_Group_Name = col_character(),
##   CD_Deaths_Reported = col_double(),
##   CD_No_of_Autopsy_conducted = col_character(),
##   CD_No_of_Cases_registered_in_connection_with_deaths = col_character(),
##   CD_No_of_Judicial_enquiry_orderedconducted = col_character(),
##   CD_No_of_Magisterial_enquiry_orderedconducted = col_character(),
##   CD_No_of_Policemen_Charge_sheeted = col_character(),
##   CD_No_of_Policemen_Convicted = col_character()
## )
```

```
head(prod)
```

Area_Name	**Y…**	**Group_Name**
<chr>	<dbl>	<chr>
Andhra Pradesh	2001	During Production/Process in Courts/Journey Connected with Investigation
Arunachal Pradesh	2001	During Production/Process in Courts/Journey Connected with Investigation
Assam	2001	During Production/Process in Courts/Journey Connected with Investigation
Bihar	2001	During Production/Process in Courts/Journey Connected with Investigation
Chandigarh	2001	During Production/Process in Courts/Journey Connected with Investigation
Chhattisgarh	2001	During Production/Process in Courts/Journey Connected with Investigation

6 rows | 1-3 of 11 columns

```
hosp<- read_csv("D:/Sem 1/PPC/A 3/40_04_Custodial_death_during_hospitalization_or_treatment.csv"
)
```

```
## Parsed with column specification:
## cols(
##   Area_Name = col_character(),
##   Year = col_double(),
##   Group_Name = col_character(),
##   Sub_Group_Name = col_character(),
##   CD_Hospitalisation_Treatment = col_double()
## )
```

```
head(hosp)
```

Area_Name	Year	Group_Name
<chr>	<dbl>	<chr>
Andhra Pradesh	2001	During Hospitalisation/Treatment/Other Reasons
Arunachal Pradesh	2001	During Hospitalisation/Treatment/Other Reasons
Bihar	2001	During Hospitalisation/Treatment/Other Reasons
Chandigarh	2001	During Hospitalisation/Treatment/Other Reasons
Chhattisgarh	2001	During Hospitalisation/Treatment/Other Reasons
Delhi	2001	During Hospitalisation/Treatment/Other Reasons

6 rows | 1-3 of 5 columns

Hosp and Prod were then merged using dplyr package’s left_join() function, on the variables Area Name and Year.

4 redundant columns, Group_Name and Sub_Group_Name of both original data sets were removed from d1, because they contained duplicate and irrelevant data.

```
d1<- hosp %>% left_join(prod,by = c("Area_Name" = "Area_Name", "Year" = "Year"))
d1<- d1[,-c(3,4,6,7)]

head(d1)
```

Area_Name	Y…	CD_Hospitalisation_Treatment	CD_Deaths_Reported	CD_No_of_Au
<chr>	<dbl>	<dbl>	<dbl>	<chr>
Andhra Pradesh	2001	15	3	3
Arunachal Pradesh	2001	1	0	0
Bihar	2001	0	0	0
Chandigarh	2001	0	0	0
Chhattisgarh	2001	0	0	0
Delhi	2001	0	*NA*	*NA*

6 rows | 1-5 of 10 columns

Understand

Inspection of d1 data frame revealed the data types of all variables using str() function. It revealed that variables 5 to 10 actually contained numeric data but were had incorrect data type of character. as.numeric() function was performed to change the data types of these 5 variables to numeric.

```
str(d1)
```

```
## tibble [213 x 10] (S3: tbl_df/tbl/data.frame)
##  $ Area_Name                                          : chr [1:213] "Andhra Pradesh" "Arunach
al Pradesh" "Bihar" "Chandigarh" ...
##  $ Year                                               : num [1:213] 2001 2001 2001 2001 2001
...
##  $ CD_Hospitalisation_Treatment                       : num [1:213] 15 1 0 0 0 0 3 1 0 0 ...
##  $ CD_Deaths_Reported                                 : num [1:213] 3 0 0 0 0 NA 1 0 0 0 ...
##  $ CD_No_of_Autopsy_conducted                         : chr [1:213] "3" "0" "0" "0" ...
##  $ CD_No_of_Cases_registered_in_connection_with_deaths: chr [1:213] "3" "0" "0" "0" ...
##  $ CD_No_of_Judicial_enquiry_orderedconducted         : chr [1:213] "1" "0" "0" "0" ...
##  $ CD_No_of_Magisterial_enquiry_orderedconducted      : chr [1:213] "1" "0" "0" "0" ...
##  $ CD_No_of_Policemen_Charge_sheeted                  : chr [1:213] "0" "0" "0" "0" ...
##  $ CD_No_of_Policemen_Convicted                       : chr [1:213] "0" "0" "0" "0" ...
```

```
d1[, c(5:10)] <- sapply(d1[, c(5:10)], as.numeric)
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

```
str(d1)
```

```
## tibble [213 x 10] (S3: tbl_df/tbl/data.frame)
##  $ Area_Name                                          : chr [1:213] "Andhra Pradesh" "Arunach
al Pradesh" "Bihar" "Chandigarh" ...
##  $ Year                                               : num [1:213] 2001 2001 2001 2001 2001
...
##  $ CD_Hospitalisation_Treatment                       : num [1:213] 15 1 0 0 0 0 3 1 0 0 ...
##  $ CD_Deaths_Reported                                 : num [1:213] 3 0 0 0 0 NA 1 0 0 0 ...
##  $ CD_No_of_Autopsy_conducted                         : num [1:213] 3 0 0 0 0 NA 1 0 0 0 ...
##  $ CD_No_of_Cases_registered_in_connection_with_deaths: num [1:213] 3 0 0 0 0 NA 1 0 0 0 ...
##  $ CD_No_of_Judicial_enquiry_orderedconducted         : num [1:213] 1 0 0 0 0 NA 0 0 0 0 ...
##  $ CD_No_of_Magisterial_enquiry_orderedconducted      : num [1:213] 1 0 0 0 0 NA 1 0 0 0 ...
##  $ CD_No_of_Policemen_Charge_sheeted                  : num [1:213] 0 0 0 0 0 NA 0 0 0 0 ...
##  $ CD_No_of_Policemen_Convicted                       : num [1:213] 0 0 0 0 0 NA 0 0 0 0 ...
```

A new variable named Location was added to d1 data set, which categorised the Area_Name(state) variable according to location/region such as Northern/Southern/Eastern etc.

It was then converted to a factor variable using as.factor() function.

```
d1$Location <- if_else(d1$Area_Name== "Chhattisgarh" |d1$Area_Name== "Madhya Pradesh",
                       "Central",
               if_else(d1$Area_Name== "Bihar" |d1$Area_Name== "Jharkhand" |d1$Area_Name== "Odish
a" |d1$Area_Name== "Sikkim" |d1$Area_Name== "West Bengal",
                       "Eastern",
               if_else(d1$Area_Name== "Arunachal Pradesh" |d1$Area_Name== "Assam"|d1$Area_Name==
"Manipur" |d1$Area_Name== "Meghalaya" |d1$Area_Name== "Mizoram"|d1$Area_Name== "Nagaland"|d1$Are
a_Name== "Tripura",
                       "Northeastern",
               if_else(d1$Area_Name== "Chandigarh" |d1$Area_Name== "Delhi" |d1$Area_Name== "Hary
ana" |d1$Area_Name== "Himachal Pradesh" |d1$Area_Name== "Jammu & Kashmir" |d1$Area_Name== "Punja
b" |d1$Area_Name== "Uttar Pradesh"|d1$Area_Name== "Uttarakhand",
                       "Northern",
               if_else(d1$Area_Name== "Andhra Pradesh"|d1$Area_Name== "Karnataka" |d1$Area_Name=
= "Kerala" |d1$Area_Name== "Tamil Nadu" |d1$Area_Name== "Telangana",
                       "Southern",
               if_else(d1$Area_Name== "Dadra & Nagar Haveli" |d1$Area_Name== "Daman & Diu" |d1$A
rea_Name== "Goa" |d1$Area_Name== "Gujarat" |d1$Area_Name== "Maharashtra" |d1$Area_Name== "Rajast
han",
                       "Western",
                       "Other"))))))

d1$Location <-as.factor(d1$Location)

summary(d1$Location)
```

```
##      Central      Eastern Northeastern     Northern     Southern 
##           18           32           47           51           33 
##      Western 
##           32 
```

Tidy & Manipulate Data I

The structure of the data table was first checked to verify whether it was tidy or not, using head() function. It revealed that the data was tidy, since observations have been plotted correctly, and rows contain individual values and not variables.

```
head(d1)
```

Area_Name <chr>	Y… <dbl>	CD_Hospitalisation_Treatment <dbl>	CD_Deaths_Reported <dbl>	CD_No_of_/
Andhra Pradesh	2001	15	3	
Arunachal Pradesh	2001	1	0	
Bihar	2001	0	0	
Chandigarh	2001	0	0	
Chhattisgarh	2001	0	0	

Area_Name <chr>	Y… <dbl>	CD_Hospitalisation_Treatment <dbl>	CD_Deaths_Reported <dbl>	CD_No_of_
Delhi	2001	0	NA	

6 rows | 1-5 of 11 columns

However, the data set needed to be reordered since the categorical variable “Location” was the last column. It was shifted to be the number two column, since it contained descriptive information about values from column one “Area_Name”. The reordering was performed using subset() function.

```
d1 <- subset(d1, select=c(1,11,2,3,4,5,6,7,8,9,10))
head(d1)
```

Area_Name <chr>	Location <fctr>	Y… <dbl>	CD_Hospitalisation_Treatment <dbl>	CD_Deaths_Reporte <dbl
Andhra Pradesh	Southern	2001	15	3
Arunachal Pradesh	Northeastern	2001	1	0
Bihar	Eastern	2001	0	0
Chandigarh	Northern	2001	0	0
Chhattisgarh	Central	2001	0	0
Delhi	Northern	2001	0	NA

6 rows | 1-5 of 11 columns

Tidy & Manipulate Data II

Next, a new variable named “Judicial_or_Magisterial_enquiry_conducted” was created which contained the sum of values between the two variables “CD_No_of_Judicial_enquiry_orderedconducted” and “CD_No_of_Magisterial_enquiry_orderedconducted”.

Mutate() function was used to create the new variable. Prior to mutating, NAs were converted to 0 in the two summed variables to prevent missing values.

The two summed variables were then dropped using subset() function to prevent redundancy.

```
d1$CD_No_of_Judicial_enquiry_orderedconducted[is.na(d1$CD_No_of_Judicial_enquiry_orderedconducte
d)] <- 0
d1$CD_No_of_Magisterial_enquiry_orderedconducted[is.na(d1$CD_No_of_Magisterial_enquiry_orderedco
nducted)] <- 0

d1<- mutate(d1,
       Judicial_or_Magisterial_enquiry_conducted = CD_No_of_Judicial_enquiry_orderedconducted +
 CD_No_of_Magisterial_enquiry_orderedconducted
       )


d1 <- subset(d1, select=c(1,2,3,4,5,6,7,10,11,12))

head(d1)
```

Area_Name <chr>	Location <fctr>	Y... <dbl>	CD_Hospitalisation_Treatment <dbl>	CD_Deaths_Reporte <dbl
Andhra Pradesh	Southern	2001	15	3
Arunachal Pradesh	Northeastern	2001	1	0
Bihar	Eastern	2001	0	0
Chandigarh	Northern	2001	0	0
Chhattisgarh	Central	2001	0	0
Delhi	Northern	2001	0	NA

6 rows | 1-5 of 10 columns

Scan I

d1 data set was scanned for missing values and errors using is.na() function and is.infinite() function.

is.na() revealed the number of NAs per column.

```
colSums(is.na(d1))  #check na
```

```
##                                           Area_Name
##                                                   0
##                                            Location
##                                                   0
##                                                Year
##                                                   0
##                        CD_Hospitalisation_Treatment
##                                                   0
##                                  CD_Deaths_Reported
##                                                  25
##                          CD_No_of_Autopsy_conducted
##                                                  28
## CD_No_of_Cases_registered_in_connection_with_deaths
##                                                  31
##                   CD_No_of_Policemen_Charge_sheeted
##                                                  37
##                        CD_No_of_Policemen_Convicted
##                                                  39
##           Judicial_or_Magisterial_enquiry_conducted
##                                                   0
```

is.infinite() function revealed that there were no infinite values in d1.

```
test<- do.call(cbind, lapply(d1, is.infinite)) #check infinite

any(test=="TRUE")
```

```
## [1] FALSE
```

To deal with high number of NAs in the data set, NAs were changed to 0 since to prevent errors in functions.

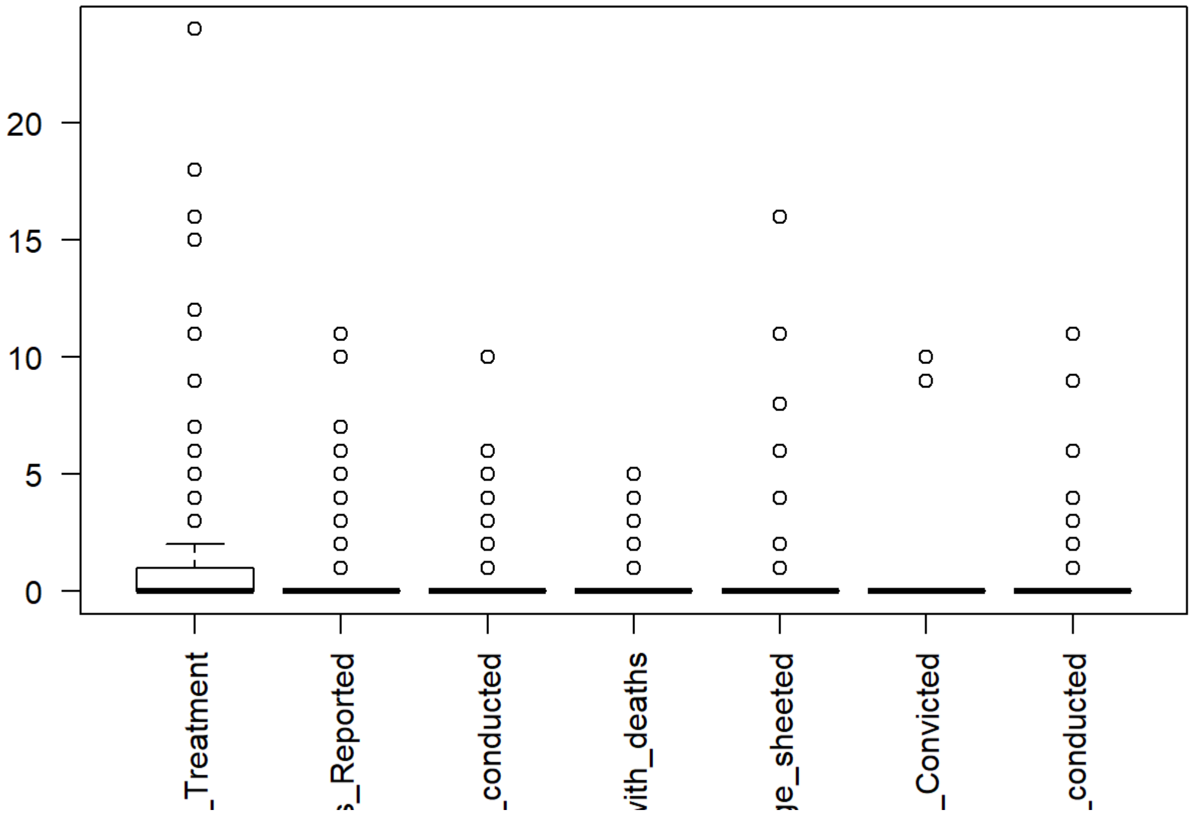
```
d1[is.na(d1)] <- 0
colSums(is.na(d1))  #check na
```

```
##                                           Area_Name
##                                                   0
##                                            Location
##                                                   0
##                                                Year
##                                                   0
##                         CD_Hospitalisation_Treatment
##                                                   0
##                                   CD_Deaths_Reported
##                                                   0
##                           CD_No_of_Autopsy_conducted
##                                                   0
## CD_No_of_Cases_registered_in_connection_with_deaths
##                                                   0
##                    CD_No_of_Policemen_Charge_sheeted
##                                                   0
##                         CD_No_of_Policemen_Convicted
##                                                   0
##            Judicial_or_Magisterial_enquiry_conducted
##                                                   0
```

Scan II

All 7 numeric variables were scanned for outliers using boxplot function. The plots revealed a large number of outliers in the majority of columns, along with a low mean.

```
boxplot(d1[,-c(1,2,3)], las=2)
```



Further examination and summary of the numeric variables revealed mean was less than 1 for 6 out of 7 variables.

Additionally, 6 variables had an Interquartile range equal to 0. This signified an issue with the data.

```
summary(d1[,-c(1,2,3)])
```

```
##  CD_Hospitalisation_Treatment CD_Deaths_Reported
##  Min.   : 0.000               Min.   : 0.0000
##  1st Qu.: 0.000               1st Qu.: 0.0000
##  Median : 0.000               Median : 0.0000
##  Mean   : 1.418               Mean   : 0.7089
##  3rd Qu.: 1.000               3rd Qu.: 0.0000
##  Max.   :24.000               Max.   :11.0000
##  CD_No_of_Autopsy_conducted
##  Min.   : 0.0000
##  1st Qu.: 0.0000
##  Median : 0.0000
##  Mean   : 0.5822
##  3rd Qu.: 0.0000
##  Max.   :10.0000
##  CD_No_of_Cases_registered_in_connection_with_deaths
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3427
##  3rd Qu.:0.0000
##  Max.   :5.0000
##  CD_No_of_Policemen_Charge_sheeted CD_No_of_Policemen_Convicted
##  Min.   : 0.0000                   Min.   : 0.0000
##  1st Qu.: 0.0000                   1st Qu.: 0.0000
##  Median : 0.0000                   Median : 0.0000
##  Mean   : 0.2488                   Mean   : 0.0892
##  3rd Qu.: 0.0000                   3rd Qu.: 0.0000
##  Max.   :16.0000                   Max.   :10.0000
##  Judicial_or_Magisterial_enquiry_conducted
##  Min.   : 0.0000
##  1st Qu.: 0.0000
##  Median : 0.0000
##  Mean   : 0.4836
##  3rd Qu.: 0.0000
##  Max.   :11.0000
```

The number of 0 values per numeric variable were calculated using colsums() function, which displayed a high number of 0s per column out of 213 observations. This implied an issue with the type of data, sample size or data collection method.

Since this data records number of deaths in specific situations per year for each Indian state, it can be ascertained that the high number of outliers is an anomaly. Moreover, the outliers cannot be excluded or imputed since that would lead to distortion of highly specific information.

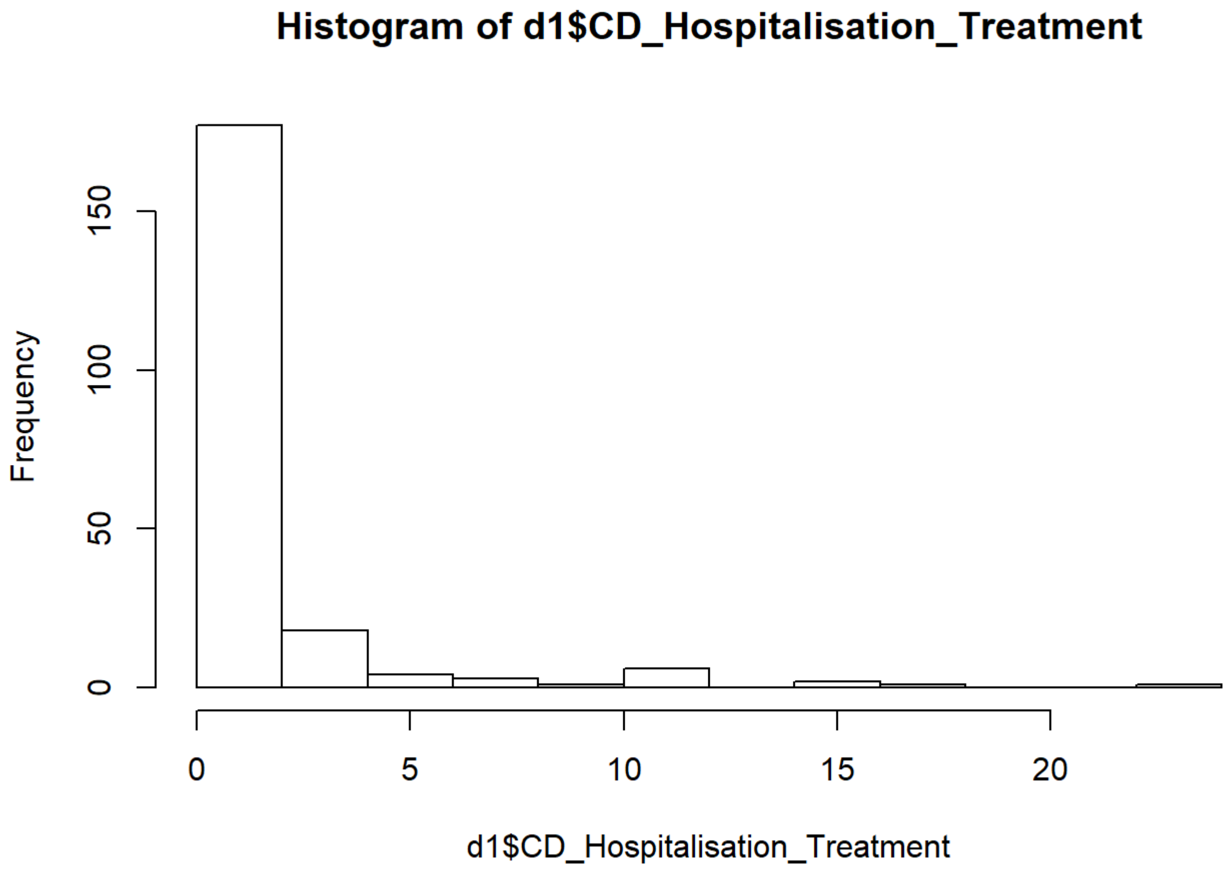
```
colSums(d1[-c(1,2,3)] == 0)
```

```
##                         CD_Hospitalisation_Treatment
##                                                  139
##                                   CD_Deaths_Reported
##                                                  160
##                           CD_No_of_Autopsy_conducted
##                                                  163
## CD_No_of_Cases_registered_in_connection_with_deaths
##                                                  174
##                    CD_No_of_Policemen_Charge_sheeted
##                                                  202
##                         CD_No_of_Policemen_Convicted
##                                                  211
##            Judicial_or_Magisterial_enquiry_conducted
##                                                  175
```

Transform

By plotting a histogram, it can be observed that the “CD_Hospitalisation_Treatment” variable has a highly right-skewed distribution.

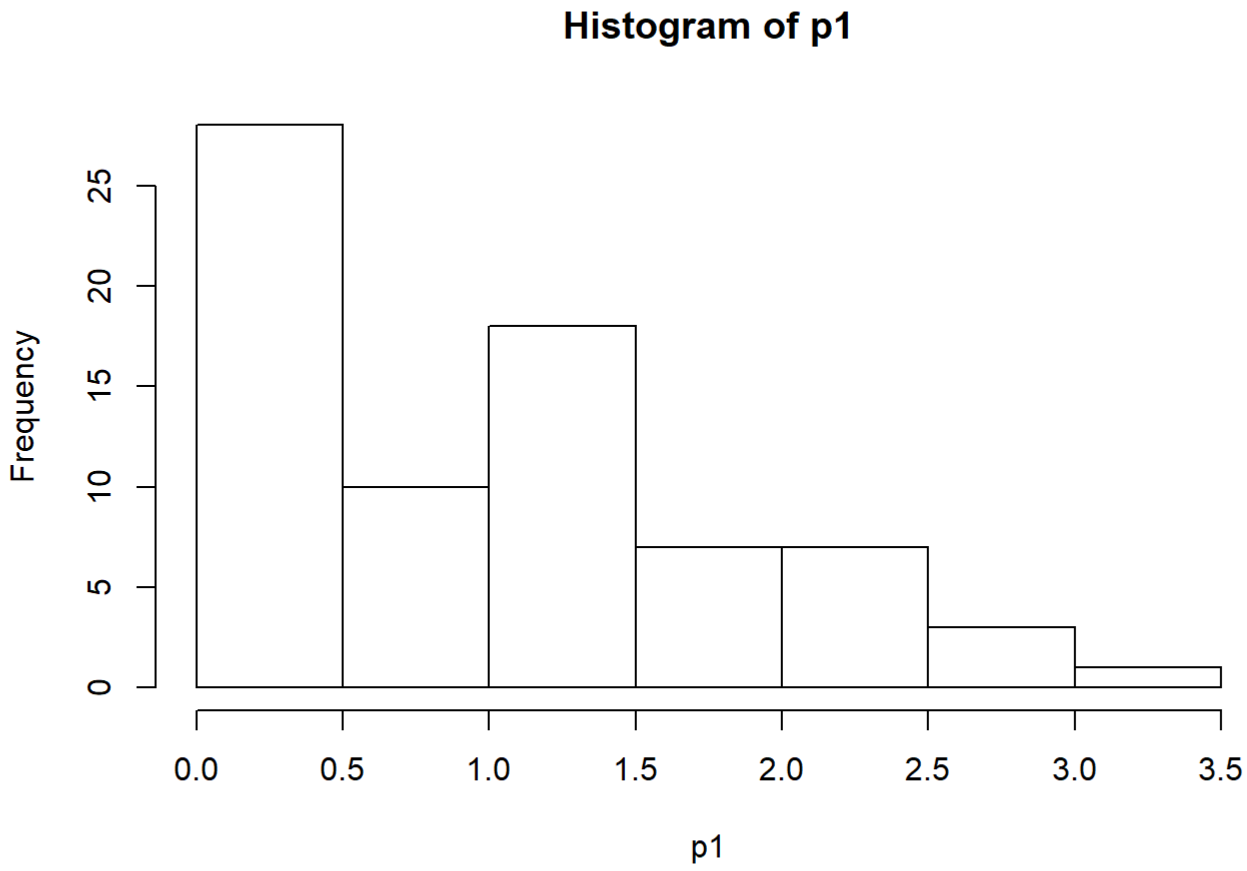
```
hist(d1$CD_Hospitalisation_Treatment)
```



Natural logarithmic transformation is applied to the variable which significantly reduces the degree of skewness. The transformation is done using the log() function.

```
p1<- log(d1$CD_Hospitalisation_Treatment)

hist(p1)
```



Conclusion

A wide and diverse range of preprocessing functions are performed on the data. Multiple R packages are explored throughout the process. The data is cleaned, wrangled and preprocessed to prepare it for further statistical modelling.