Executive Summary:

1. Box plot was used to check the outliers in the data
2. Data distribution was checked to figure if distribution is normal so that appropriate method can be chosen to test outliers (IQR/z-score)
3. Covid period was identified as “2020-03-21” to “2021-04-10” from the IQR method
4. Cubic spline methodology was used to impute values that follow non-linear trend in the data (i.e. covid period values) and lambda value was chosen as 0.11 (a moderate value was chosen but not higher value so that overfitting doesn’t occur)
5. MAE and MPE should ideally be close to zero and it holds good for additive model rather than the multiplicative one - hence the point forecast of additive one might be close to the original value

```
#Load the required Libraries

if (!require("imputeTS")) install.packages("imputeTS")
```

```
## Loading required package: imputeTS
```

```
## Warning: package 'imputeTS' was built under R version 4.3.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(fredr)
```

```
## Warning: package 'fredr' was built under R version 4.3.2
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.3.2
```

```
library(reprex)
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.3.2
```

```
##
## Attaching package: 'zoo'
```

```
## The following object is masked from 'package:imputeTS':
##
##     na.locf
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(imputeTS)
library(forecast)
```

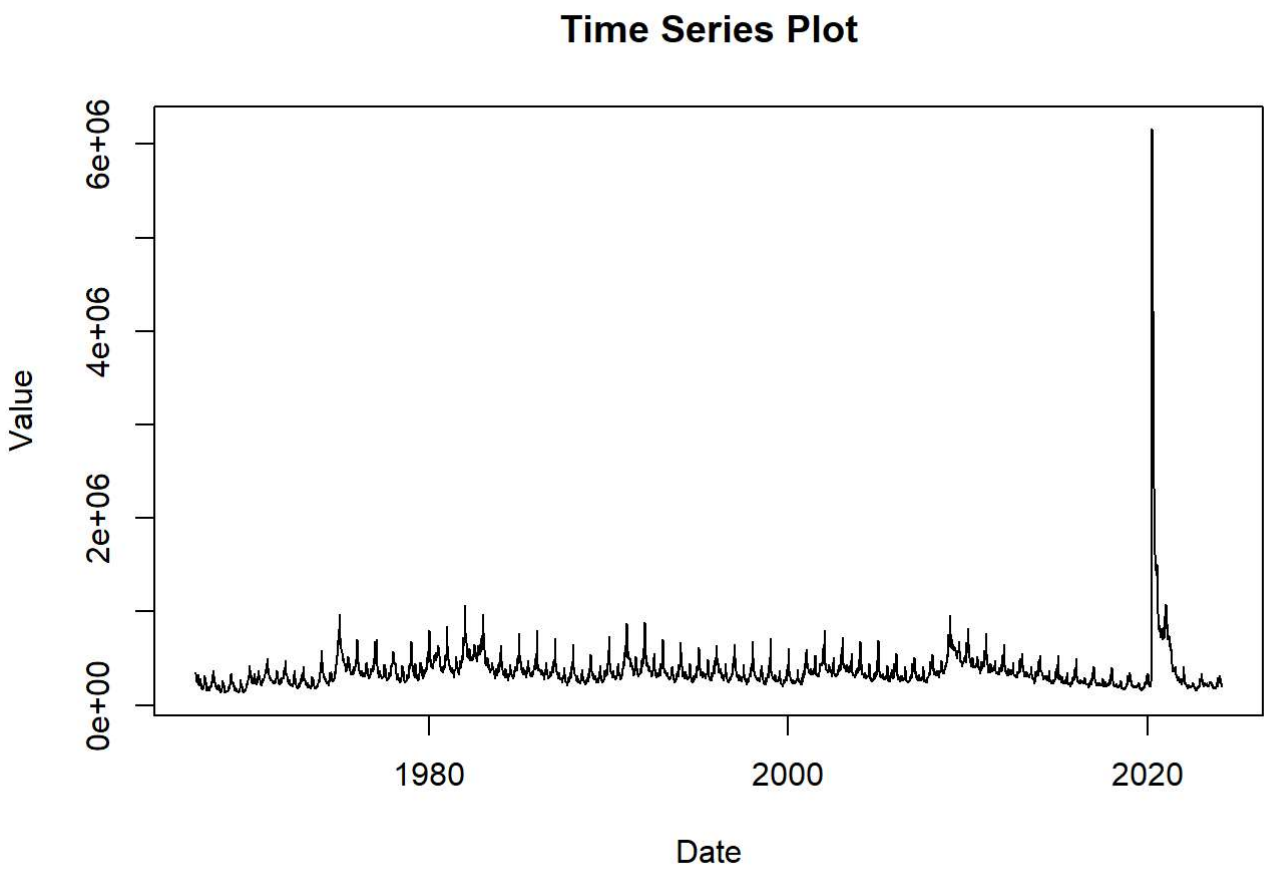
```
#Load the data and view the sample entries

fredr_set_key("e3dbddf1961174f1b64d469145d6eff9")
icnsa_data = fredr(series_id = "ICNSA")
head(icnsa_data)
```

```
## # A tibble: 6 × 5
##   date       series_id  value realtime_start realtime_end
##   <date>     <chr>      <dbl> <date>         <date>
## 1 1967-01-07 ICNSA     346000 2024-02-28     2024-02-28
## 2 1967-01-14 ICNSA     334000 2024-02-28     2024-02-28
## 3 1967-01-21 ICNSA     277000 2024-02-28     2024-02-28
## 4 1967-01-28 ICNSA     252000 2024-02-28     2024-02-28
## 5 1967-02-04 ICNSA     274000 2024-02-28     2024-02-28
## 6 1967-02-11 ICNSA     276000 2024-02-28     2024-02-28
```

```
#Drop insignificant columns

icnsa_data <- icnsa_data[, c("date", "value")]
head(icnsa_data)
```

```
## # A tibble: 6 × 2
##   date        value
##   <date>      <dbl>
## 1 1967-01-07 346000
## 2 1967-01-14 334000
## 3 1967-01-21 277000
## 4 1967-01-28 252000
## 5 1967-02-04 274000
## 6 1967-02-11 276000
```

```
#Time Series plot of ICNSA data
plot(icnsa_data$date, icnsa_data$value, type = "l", xlab = "Date", ylab = "Value", main = "Time
Series Plot")
```



Exploratory Data Analysis

```
#Displays number of rows and columns in the dataset
dim(icnsa_data)
```

```
## [1] 2981    2
```

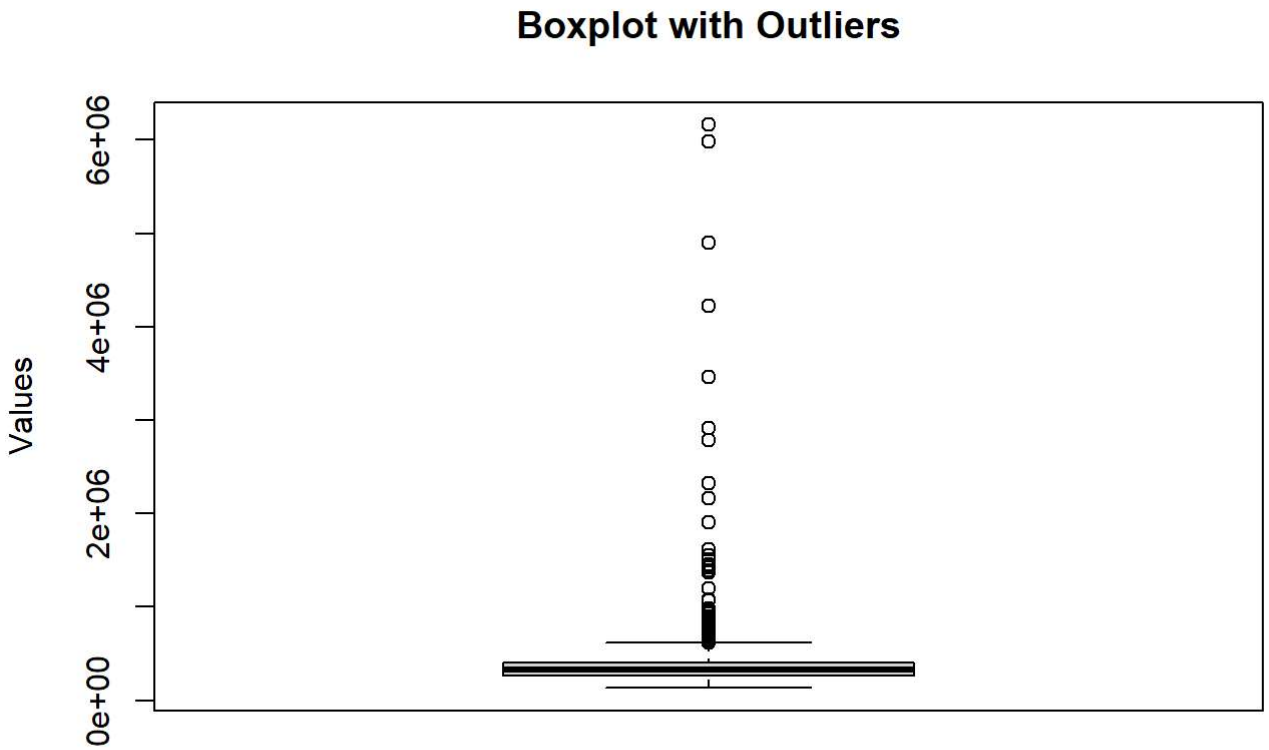
```
#Datatype of each column of the Iris Dataset
str(icnsa_data)
```

```
## tibble [2,981 × 2] (S3: tbl_df/tbl/data.frame)
##  $ date : Date[1:2981], format: "1967-01-07" "1967-01-14" ...
##  $ value: num [1:2981] 346000 334000 277000 252000 274000 276000 247000 248000 326000 240000
...
```

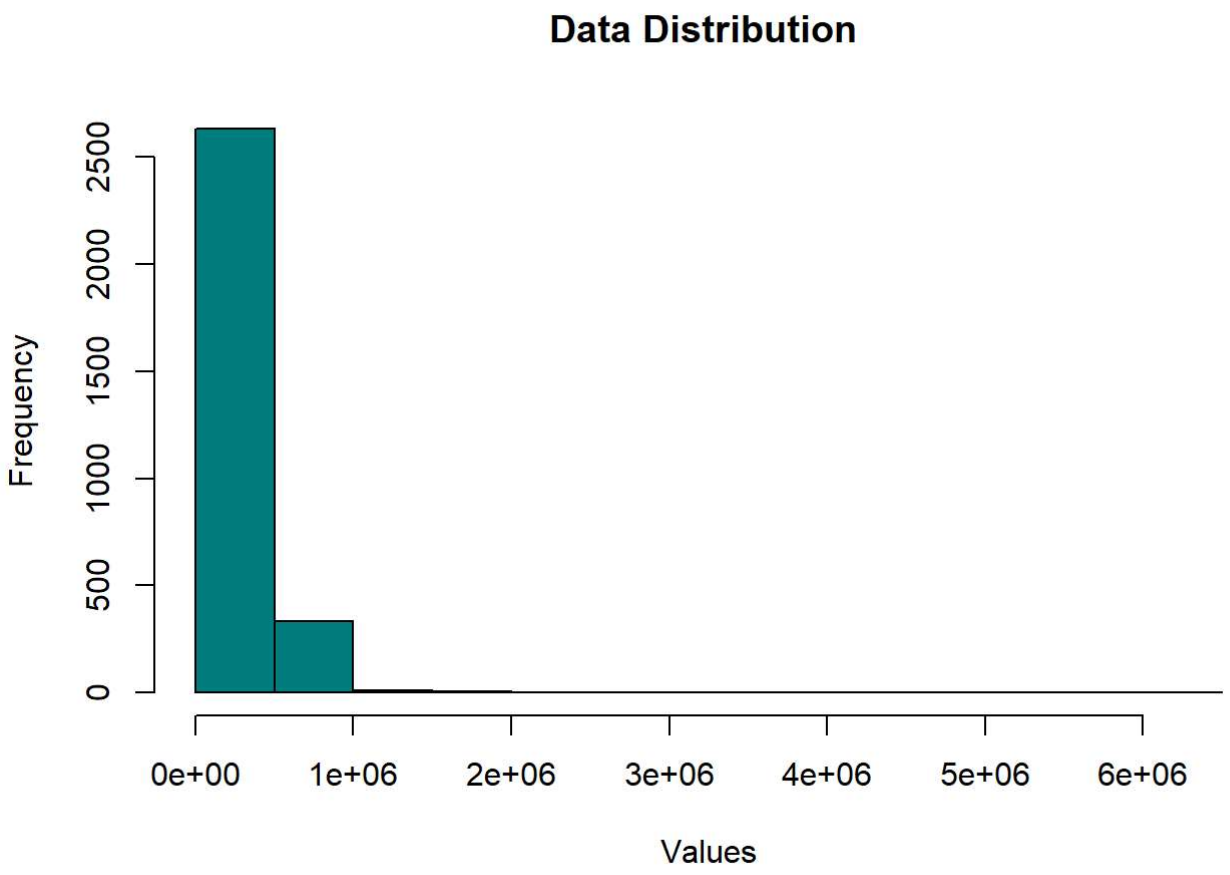
```
#Count number of null values in each column - No null values found in any column

Icnsa_NullValues<-as.data.frame(sapply(icnsa_data, function(x) sum(is.na(x))))
colnames(Icnsa_NullValues) <- c('Null Value Count')
Icnsa_NullValues
```

```
##       Null Value Count
## date                 0
## value                0
```

```
#Box-plot to check the outliers
boxplot(icnsa_data['value'], main = "Boxplot with Outliers", ylab = "Values")
```




```
#Histogram to check the distribution of the data
hist(icnsa_data$value, main = "Data Distribution ", xlab = "Values", ylab = "Frequency",col="#00
8080")
```



#Since the data doesn't have normal distribution, we have to use IQR to find the outliers in the data instead of Z-score

```
#IQR for calculating outliers range in the data

data <- icnsa_data$value

Q1 <- quantile(data, 0.25)
Q3 <- quantile(data, 0.75)

# Calculate IQR
IQR <- Q3 - Q1

#Threshold Multiplier
threshold <- 1.5

# Calculate lower and upper bounds for outliers
lower_bound <- Q1 - threshold * IQR
upper_bound <- Q3 + threshold * IQR

# Identify outliers
outliers <- data[data < lower_bound | data > upper_bound]

print(outliers)
```

```
##   [1]  693000  637000  677000  813000  681000  969000  850000  729000  699000
##  [10]  691000  708000  685000  704000  680000  804000  648000  643000  628000
##  [19]  839000  638000  726000  657000  644000  702000  694300 1073500  761700
##  [28]  771200  692300  671000  631400  647000  638100  650400  709400  638200
##  [37]  653600  745100  976600  773600  650600  621600  637900  770000  803900
##  [46]  710493  654620  730995  637449  649471  651775  872742  691092  625772
##  [55]  652046  882118  687914  704930  676076  637910  654473  682016  713805
##  [64]  647045  637343  799246  620929  620004  724111  677897  693776  760481
##  [73]  629867  719691  717000  731958  956791  763987  620143  682176  710152
##  [82]  619951  645827  652635  623279  677038  673097  651215  825891  659173
##  [91]  773499  646219 2914268 5981838 6161268 4898119 4221704 3466665 2790860
## [100] 2325889 2162822 1902006 1610909 1555824 1456353 1446176 1425969 1390315
## [109] 1511632 1372247 1200962  982808  830266  878555  813744  826436  857006
## [118]  784765  814441  731391  722387  820590  757745  731660  736618  719716
## [127]  708717  827447  710901  946091  928178  854113  823512  898597 1082696
## [136]  930184  833704  843702  860209  827835  707268  749251  714661  750287
## [145]  630621  701073  668346  620420
```

```
print(lower_bound)
```

```
##     25%
## 54535.5
```

```
print(upper_bound)
```

```
##      75%
## 617891.5
```

```
#Print the dates for which outliers exist and filter the data post 2020 to get covid period

class(icnsa_data$date)
```

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

```
outlier_dates <- c()
for (value in outliers) {
  outlier_rows <- which(icnsa_data$value == value)
  outlier_date <- icnsa_data$date[outlier_rows]
  if (as.numeric(format(outlier_date, "%Y")) >= 2020){
  print(outlier_date)

  }
}
```

```
## [1] "2020-03-21"
## [1] "2020-03-28"
## [1] "2020-04-04"
## [1] "2020-04-11"
## [1] "2020-04-18"
## [1] "2020-04-25"
## [1] "2020-05-02"
## [1] "2020-05-09"
## [1] "2020-05-16"
## [1] "2020-05-23"
## [1] "2020-05-30"
## [1] "2020-06-06"
## [1] "2020-06-13"
## [1] "2020-06-20"
## [1] "2020-06-27"
## [1] "2020-07-04"
## [1] "2020-07-11"
## [1] "2020-07-18"
## [1] "2020-07-25"
## [1] "2020-08-01"
## [1] "2020-08-08"
## [1] "2020-08-15"
## [1] "2020-08-22"
## [1] "2020-08-29"
## [1] "2020-09-05"
## [1] "2020-09-12"
## [1] "2020-09-19"
## [1] "2020-09-26"
## [1] "2020-10-03"
## [1] "2020-10-10"
## [1] "2020-10-17"
## [1] "2020-10-24"
## [1] "2020-10-31"
## [1] "2020-11-07"
## [1] "2020-11-14"
## [1] "2020-11-21"
## [1] "2020-11-28"
## [1] "2020-12-05"
## [1] "2020-12-12"
## [1] "2020-12-19"
## [1] "2020-12-26"
## [1] "2021-01-02"
## [1] "2021-01-09"
## [1] "2021-01-16"
## [1] "2021-01-23"
## [1] "2021-01-30"
## [1] "2021-02-06"
## [1] "2021-02-13"
## [1] "2021-02-20"
## [1] "2021-02-27"
## [1] "2021-03-06"
## [1] "2021-03-13"
```

```
## [1] "2021-03-20"
## [1] "2021-03-27"
## [1] "2021-04-03"
## [1] "2021-04-10"
```

#Observation: 1. Therefore, “2020-03-21” to “2021-04-10” is the covid period from the above data

3. Use a cubic spline methodology to impute new values for the Covid period you decided on in #2. Discuss the λ value chosen and the overall fit.

```
#Define Covid period

covid_start <- as.Date("2020-03-21")
covid_end <- as.Date("2021-04-10")

# Filter the relevant COVID period
covid_data  <- icnsa_data[icnsa_data$date >= covid_start & icnsa_data$date <= covid_end, ]

# Define lambda value
lambda <- 0.11

# Fit a smooth spline to the data
smooth_fit <- smooth.spline(covid_data$date, covid_data$value, df = 3, lambda = lambda)

# Interpolate missing values using the fitted spline
imputed_values <- predict(smooth_fit, xout = covid_data$date)$y

# Replace missing values in the COVID data with the imputed values
covid_data$value <- ifelse(is.na(covid_data$value), imputed_values, covid_data$value)

# Update the original ICNSA data with the imputed COVID values
icnsa_data[icnsa_data$date >= covid_start & icnsa_data$date <= covid_end, ] <- covid_data
```

#Note: Lambda value is chosen to be 0.11 as it offers moderate level of smoothing while retaining important features - Higher values of lambda can cause over smoothing and could lead to overfitting

Fit multiplicative holt-winters multiplicative and

additive models to get the forecast

```
ts_data <- ts(icnsa_data$value, frequency = 12)

#Multiplicative Holt-Winters model
hw_multiplicative <- HoltWinters(ts_data, seasonal = "multiplicative")
forecast_multiplicative <- forecast(hw_multiplicative, h = 1)

#Additive Holt-Winters model
hw_additive <- HoltWinters(ts_data, seasonal = "additive")
```

```
## Warning in HoltWinters(ts_data, seasonal = "additive"): optimization
## difficulties: ERROR: ABNORMAL_TERMINATION_IN_LNSRCH
```

```
forecast_additive <- forecast(hw_additive, h = 1)

#Print the forecasts
print("Multiplicative Holt-Winters Forecast:")
```

```
## [1] "Multiplicative Holt-Winters Forecast:"
```

```
print(forecast_multiplicative)
```

```
##         Point Forecast    Lo 80  Hi 80     Lo 95    Hi 95
## Jun 249       200155.6 43329.21 356982 -39689.68 440000.9
```

```
print("Additive Holt-Winters Forecast:")
```

```
## [1] "Additive Holt-Winters Forecast:"
```

```
print(forecast_additive)
```

```
##         Point Forecast    Lo 80    Hi 80     Lo 95    Hi 95
## Jun 249       205665.1 69306.32 342023.8 -2877.635 414207.8
```

#Observations: Point Forecast for multiplicative model: 200368.8 Point forecast for additive model: 174680.9

```
# Calculate accuracy measures
accuracy_multiplicative <- accuracy(forecast_multiplicative)
accuracy_additive <- accuracy(forecast_additive)

print("Accuracy Measures for Multiplicative Holt-Winters Forecast:")
```

```
## [1] "Accuracy Measures for Multiplicative Holt-Winters Forecast:"
```

```
print(accuracy_multiplicative)
```

```
##                  ME     RMSE      MAE        MPE     MAPE      MASE      ACF1
## Training set 631.6 122353.3 61888.31 -0.4217838 15.96123 0.5490522 0.1873774
```

```
print("Accuracy Measures for Additive Holt-Winters Forecast:")
```

```
## [1] "Accuracy Measures for Additive Holt-Winters Forecast:"
```

```
print(accuracy_additive)
```

```
##                    ME     RMSE      MAE        MPE     MAPE      MASE      ACF1
## Training set 1591.843 106395.3 53792.43 -0.2550679 14.68436 0.4772283 0.2432645
```