

# *TIME SERIES IN R- TASK1*

CHARLES WIREDU

3/3/2022

```
##CRIME DATA IN CHICAGO RELATED LAB ASSIGNMENT
```

```
##Setting work directory
```

```
setwd("~/Desktop/Data science Assignments")
```

```
##Attaching the necessary library
```

```
#install.packages("tidyr")
```

```
#install.packages("tidyverse")
```

```
#install.packages("lubridate")
```

```
#install.packages("ggplot2")
```

```
#install.packages("plotly")
```

```
#install.packages("TSstudio")
```

```
#install.packages("forecast")
```

```
#install.packages("MLmetrics")
```

```
#install.packages("dplyr")
```

```
#install.packages("viridisLite")
```

```
#install.packages("viridis")
```

```
#install.packages("padr")
```

```
#install.packages("rmdformats")
```

```
#install.packages("recipes")
```

```
library(recipes)
```

```
## Loading required package: dplyr
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```

##      filter, lag
## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union
##
## Attaching package: 'recipes'
## The following object is masked from 'package:stats':
##
##      step
library(tidyr)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --
## v ggplot2 3.3.5      v purrr 0.3.4
## v tibble 3.1.6       v stringr 1.4.0
## v readr 2.1.1        v forcats 0.5.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x stringr::fixed() masks recipes::fixed()
## x dplyr::lag() masks stats::lag()
library(lubridate)

##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:base':
##
##      date, intersect, setdiff, union
library(ggplot2)
library(plotly)

##
## Attaching package: 'plotly'
## The following object is masked from 'package:ggplot2':
##
##      last_plot
## The following object is masked from 'package:stats':
##
##      filter
## The following object is masked from 'package:graphics':
##
##      layout
library(TSstudio)
library(forecast)

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

```

```
library(MLmetrics)
```

```
##
## Attaching package: 'MLmetrics'
## The following object is masked from 'package:base':
##
##      Recall
library(dplyr)
library(viridisLite)
library(viridis)
library(padr)
```

## LOADING DATASET

```
library(readr)
Fouryears_all <- read_csv("~/Downloads/Fouryears_all.csv")

## Rows: 958000 Columns: 4
## -- Column specification -----
## Delimiter: ","
## chr  (1): Primary.Type
## dbl  (2): Latitude, Longitude
## dtm  (1): Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
View(Fouryears_all)
```

```
## Warning in system2("/usr/bin/otool", c("-L", shQuote(DSO)), stdout = TRUE):
## running command ''/usr/bin/otool' -L '/Library/Frameworks/R.framework/Resources/
## modules/R_de.so'' had status 1
```

```
crime<- Fouryears_all
```

```
glimpse(crime)
```

```
## Rows: 958,000
## Columns: 4
## $ Date      <dtm> 2017-09-01 05:00:00, 2017-09-01 05:00:00, 2017-09-01 05:~
## $ Primary.Type <chr> "THEFT", "CRIMINAL DAMAGE", "ROBBERY", "ROBBERY", "BATTER~
## $ Latitude    <dbl> 41.79859, 41.71380, 41.97409, 41.97860, 41.98563, 41.7814~
## $ Longitude   <dbl> -87.66714, -87.62727, -87.65635, -87.65993, -87.71383, -8~
```

## DATA DESCRIPTION

This data set reflects the four years of all the crimes recorded in Chicago from 2001 to present. The data has four columns and 958000 rows. Below is a summary of the data set used for this lab work. —

```
summary(crime)
```

##	Date	Primary.Type	Latitude
## Min.	:2017-09-01 05:00:00	Length:958000	Min. :36.62
## 1st Qu.	:2018-07-28 02:00:00	Class :character	1st Qu.:41.77
## Median	:2019-06-25 16:00:00	Mode :character	Median :41.86
## Mean	:2019-07-16 16:28:47		Mean :41.84
## 3rd Qu.	:2020-06-28 08:00:00		3rd Qu.:41.91

```
## Max.      :2021-08-27 23:00:00      Max.      :42.02
##                                                  NA's      :12314
## Longitude
## Min.      :-91.69
## 1st Qu.   :-87.71
## Median    :-87.66
## Mean      :-87.67
## 3rd Qu.   :-87.63
## Max.      :-87.53
## NA's      :12314
```

```
##subset dataset
## data wrangling
crime %>%
  select(Date, Primary.Type, Latitude ) %>%
  mutate( ymd_hms(Date),
          Primary.Type=as.factor(Primary.Type)) %>%
  mutate(Date = floor_date(Date, unit = "hours")) %>%
  arrange(Date) %>%
  filter(Date > "2019-09-01" & Date< "2021-09-01")
```

```
## # A tibble: 425,193 x 4
##   Date                Primary.Type    Latitude `ymd_hms(Date)`
##   <dtm>                <fct>         <dbl> <dtm>
## 1 2019-09-01 05:00:00 CRIMINAL DAMAGE    41.9 2019-09-01 05:00:00
## 2 2019-09-01 05:00:00 THEFT                41.9 2019-09-01 05:00:00
## 3 2019-09-01 05:00:00 BATTERY               41.8 2019-09-01 05:00:00
## 4 2019-09-01 05:00:00 BATTERY               41.8 2019-09-01 05:00:00
## 5 2019-09-01 05:00:00 ROBBERY               41.7 2019-09-01 05:00:00
## 6 2019-09-01 05:00:00 ROBBERY               41.7 2019-09-01 05:00:00
## 7 2019-09-01 05:00:00 BATTERY               41.8 2019-09-01 05:00:00
## 8 2019-09-01 05:00:00 CRIMINAL DAMAGE    41.9 2019-09-01 05:00:00
## 9 2019-09-01 05:00:00 THEFT                41.9 2019-09-01 05:00:00
## 10 2019-09-01 05:00:00 THEFT                41.9 2019-09-01 05:00:00
## # ... with 425,183 more rows
```

```
## cleaning the dataset
crime_clean <- crime %>%
  select(Date, Primary.Type ) %>%
  mutate(ymd_hms(Date),
          Primary.Type= as.factor(Primary.Type))%>%
  mutate(Date = floor_date(Date, unit= "hours")) %>%
  arrange(Date)
```

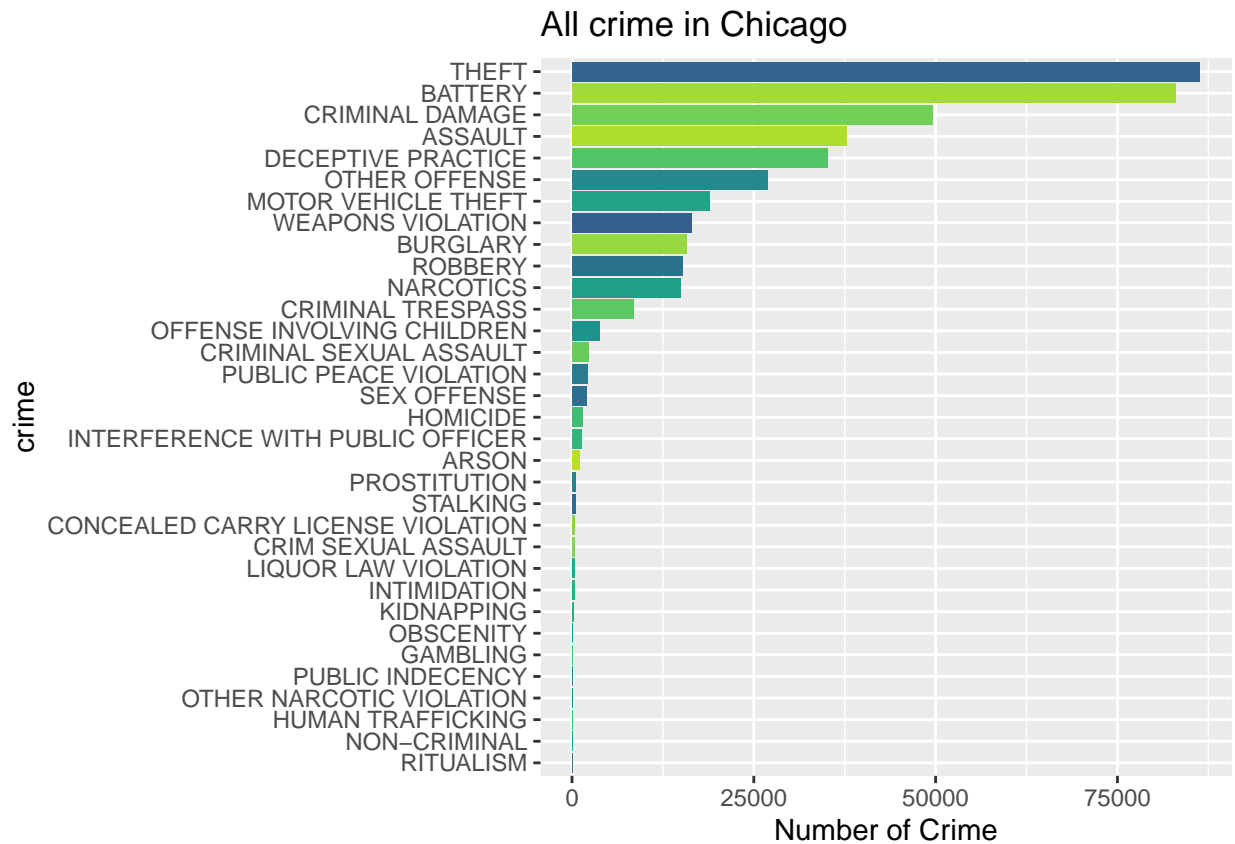
```
## removing unwanted column
crime_clean$`ymd_hms(Date)` <- NULL
## filtering two years dataset
new_date =crime_clean %>%
  select(Date, Primary.Type) %>%
  filter(Date > "2019-09-01" & Date< "2021-09-01")
```

```
## conducting data aggregation and plots
##1
new_date %>%
  group_by(Primary.Type) %>%
  summarise(count = n()) %>%
```

```

arrange(count) %>%
ggplot(aes(x= count, y= reorder(Primary.Type, count))) +
geom_col(aes(fill= Primary.Type), show.legend = F) +
scale_fill_viridis(option = "D", discrete = T, direction= 1, begin= 0.9, end= 0.3) +
labs(title= "All crime in Chicago", x= "Number of Crime", y="crime") +
theme_gray()

```



```

##2 interactive plot
plot1 <-new_date %>%
  group_by(Primary.Type) %>%
  summarise(count = n()) %>%
  arrange(count) %>%
  head(10) %>%
  ggplot(aes(x= count, y= reorder(Primary.Type, count),text= paste("new_date: ",Primary.Type,
                                                                    "<br>", "Happened: ",
                                                                    count, "times"))) +

  geom_col(aes(fill= Primary.Type), show.legend = F) +
  scale_fill_viridis(option = "D", discrete = T, direction= 1, begin= 0.9, end= 0.3) +
  labs(title= "Top 10 crime in chicago", x= "Number of Crime", y="crime") +
  theme_gray()

###summarise() ungrouping output (overriding with groups arguments)
##ggplotly(plot1,tooltip= "text")

##creating prediction time frame
crime_theft <- new_date %>%

```

```

filter(Primary.Type == 'THEFT') %>%
group_by(Date) %>%
summarise(Theft = n()) %>%
filter(Date > "2019-09-01" & Date <"2021-09-01")

##checking the last 10 of the dataset( beginning and end of the dataframe)
tail(crime_theft,10)

```

```

## # A tibble: 10 x 2
##   Date           Theft
##   <dtm>         <int>
## 1 2021-08-27 14:00:00     6
## 2 2021-08-27 15:00:00    10
## 3 2021-08-27 16:00:00     4
## 4 2021-08-27 17:00:00    12
## 5 2021-08-27 18:00:00     8
## 6 2021-08-27 19:00:00     9
## 7 2021-08-27 20:00:00     9
## 8 2021-08-27 21:00:00     4
## 9 2021-08-27 22:00:00     4
## 10 2021-08-27 23:00:00     5

```

```

crime_theft <- new_date %>%
  filter(Primary.Type == 'THEFT') %>%
  group_by(Date) %>%
  summarise(Theft = n()) %>%
  filter(Date > "2019-09-01" & Date <"2021-09-01")%>%
  slice(-c(1:4)) %>%
  ungroup()

tail(crime_theft,20)

```

```

## # A tibble: 20 x 2
##   Date           Theft
##   <dtm>         <int>
## 1 2021-08-27 03:00:00     4
## 2 2021-08-27 04:00:00     1
## 3 2021-08-27 05:00:00     6
## 4 2021-08-27 07:00:00     4
## 5 2021-08-27 08:00:00     6
## 6 2021-08-27 09:00:00     4
## 7 2021-08-27 10:00:00     3
## 8 2021-08-27 11:00:00     7
## 9 2021-08-27 12:00:00     6
## 10 2021-08-27 13:00:00     2
## 11 2021-08-27 14:00:00     6
## 12 2021-08-27 15:00:00    10
## 13 2021-08-27 16:00:00     4
## 14 2021-08-27 17:00:00    12
## 15 2021-08-27 18:00:00     8
## 16 2021-08-27 19:00:00     9
## 17 2021-08-27 20:00:00     9
## 18 2021-08-27 21:00:00     4
## 19 2021-08-27 22:00:00     4

```

```
## 20 2021-08-27 23:00:00      5
```

```
tail(crime_theft,5)
```

```
## # A tibble: 5 x 2
```

```
##   Date      Theft
```

```
##   <dtm>      <int>
```

```
## 1 2021-08-27 19:00:00      9
```

```
## 2 2021-08-27 20:00:00      9
```

```
## 3 2021-08-27 21:00:00      4
```

```
## 4 2021-08-27 22:00:00      4
```

```
## 5 2021-08-27 23:00:00      5
```

```
##after our time frame data is correct
```

```
## we check for an missing values and interval in our data
```

```
anyNA(crime_theft)
```

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

```
## we will split our train data into two type of data set names; train and test
```

```
theft_train <- head(crime_theft, nrow(crime_theft)- 365)
```

```
theft_test <- tail(crime_theft, 365)
```

```
## creating a time series object
```

```
##to creat a time series model, we need to create a time series object from our train data
```

```
theft_ts <- ts(theft_train$Theft, frequency= 24)
```

```
#decompose 1
```

```
theft_ts_dec <- theft_ts %>%
```

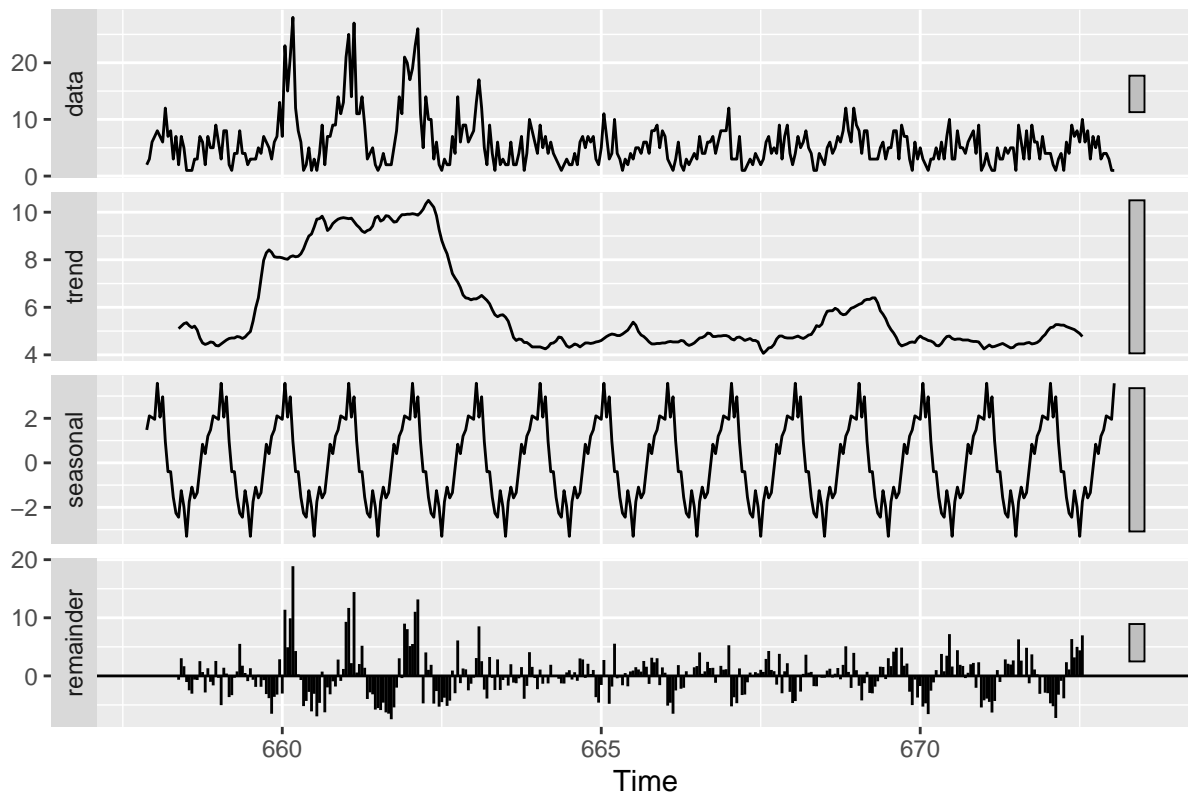
```
  tail(365) %>%
```

```
  decompose()
```

```
theft_ts_dec %>%
```

```
  autoplot()
```

## Decomposition of additive time series



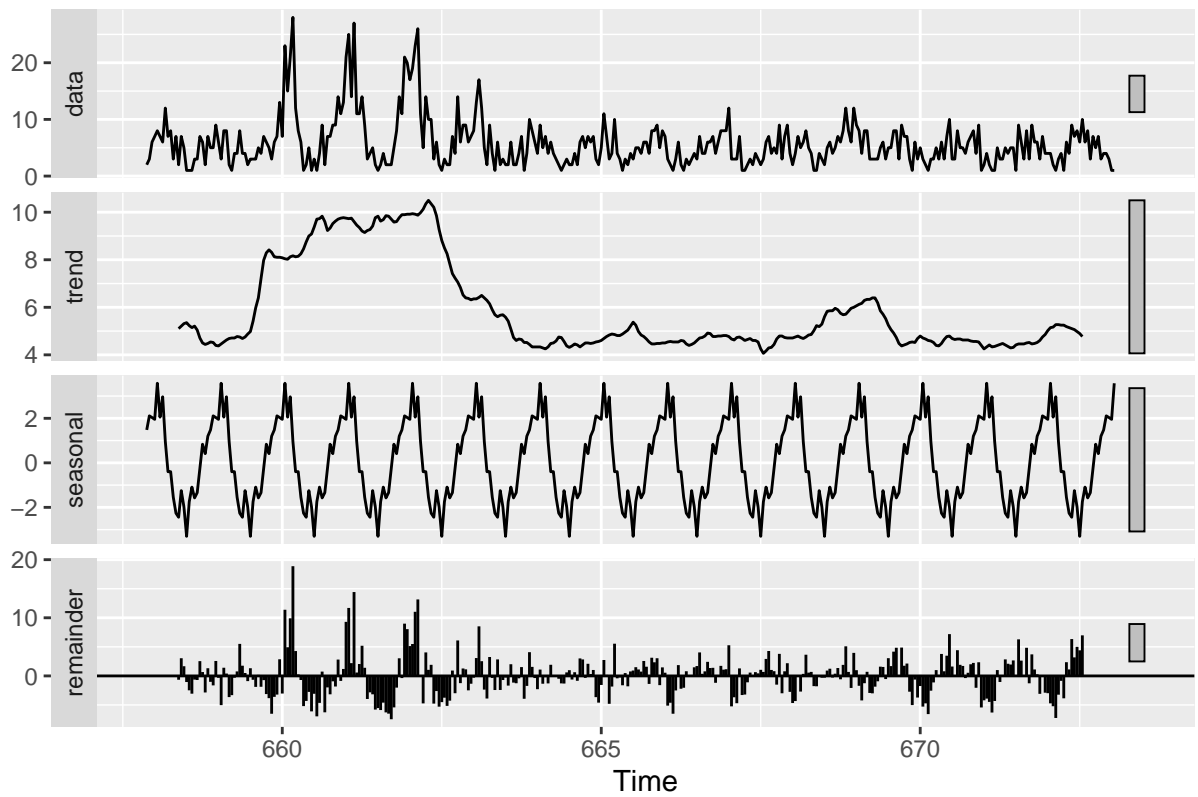
```
##create MSTs object (multi-seasonal Time Series)
Theft_multi <- msts(theft_train$Theft, seasonal.periods= c(24, #daily
                                                         24*7, # weeeekly
                                                         24*30))##monthly

## Decompose MSTs object
Theft_multi_dec <- Theft_multi %>%
  mstl()
view(Theft_multi_dec)

# 2019-2021
##MSTS OBJECT()
##par(mfrow=c(2,1))
theft_ts_dec %>%
  autoplot()
```

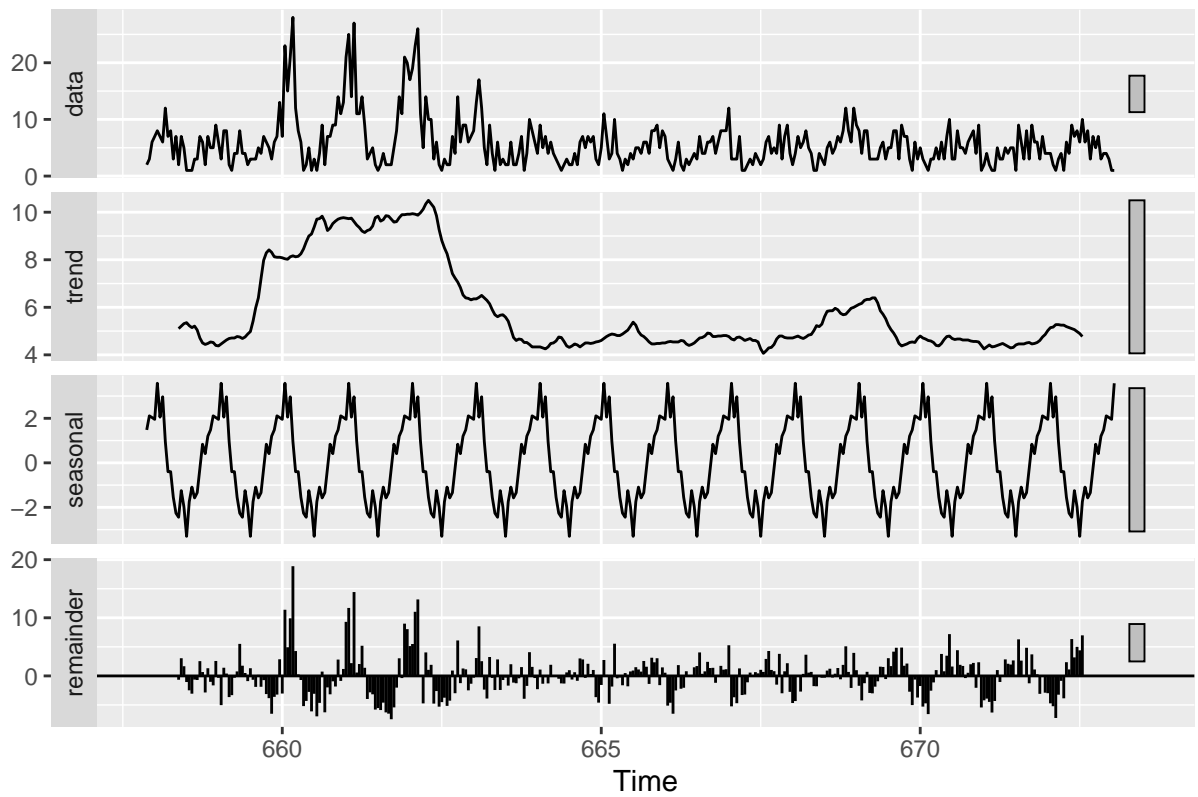


## Decomposition of additive time series

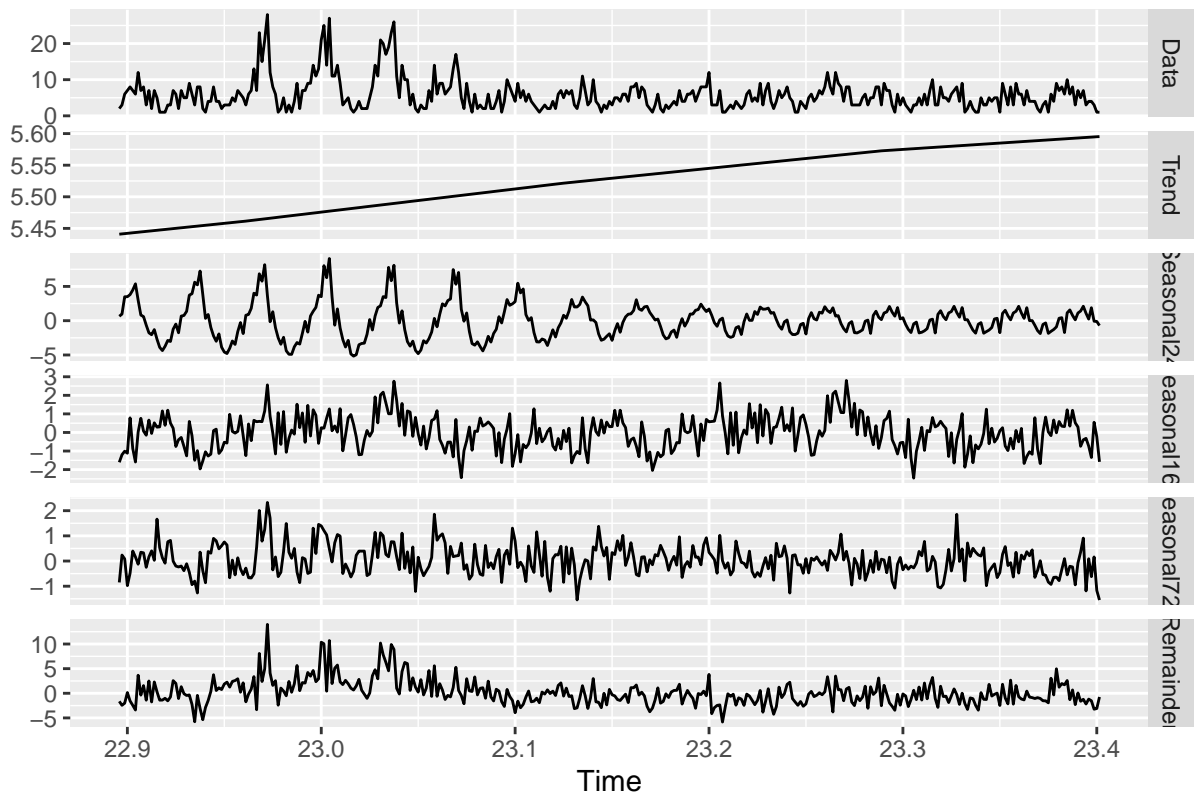


```
##Original time series object  
theft_ts_dec %>%  
  autoplot()
```

## Decomposition of additive time series



```
Theft_multi_dec %>%  
  tail(365) %>%  
  autoplot()
```



```
##Original time series object
x11()
```

```
## Warning in system2("otool", c("-L", shQuote(DSO)), stdout = TRUE): running
## command ''otool' -L '/Library/Frameworks/R.framework/Resources/modules/
## R_X11.so'' had status 1
```

```
theft_ts_dec %>%
  autoplot()
```

Decomposition of additive time series

