Handling Temporal Data with R-Programming

Adam Quek

2025-07-23

Temporal Data

Temporal data refers to observations collected over time - commonly in healthcare, economics, environmental monitoring or operational research. What distinguishes temporal data from other forms is the **time-depence** between observations.

For example, a patient's heart rate recorded at 10:00 a.m. is not independent of their reading at 10.15 a.m. This violates the classical assumption of independence in many statistical models, and necessitates time-aware techniques.

Understanding and handling temporal data requires us to think in terms of **structure over time**, particularly:

- **Lag**: How past values influence current outcomes.
- **Seasonality**: Patterns that repeat at regular intervals (daily, weekly, annually).

In this tutorial, we will walk through:

1. Loading and cleaning of temporal data
2. Extracting and working with time components
3. Detecting lags and periodicity
4. Visualising and decomposing temporal trends
5. Basic forecasting

1. Loading a Sample Dataset

In this example, we will download daily climate data from the Ogimet server through `climate` package in R. Ogimet includes stations globally (station list).

For Singapore, two stations are available:

- **48694**: Singapore/ Paya Lebar (1.37, 103.92)
- **48698**: Singapore Changi Airport (1.37, 103.98)

We will retrieve data from **Changi Airport (station 48698)** for the last five years.

```
library(climate)
changi <- meteo_ogimet(date = c(as.Date("2020-01-01"), as.Date("2025-01-01")),
                       station = 48698, interval="daily")
```

To avoid long loading times and potential server issues, we will use pre-downloaded data saved as `changi.csv`.

```
# loading and previewing climate data

library(tidyverse)
```

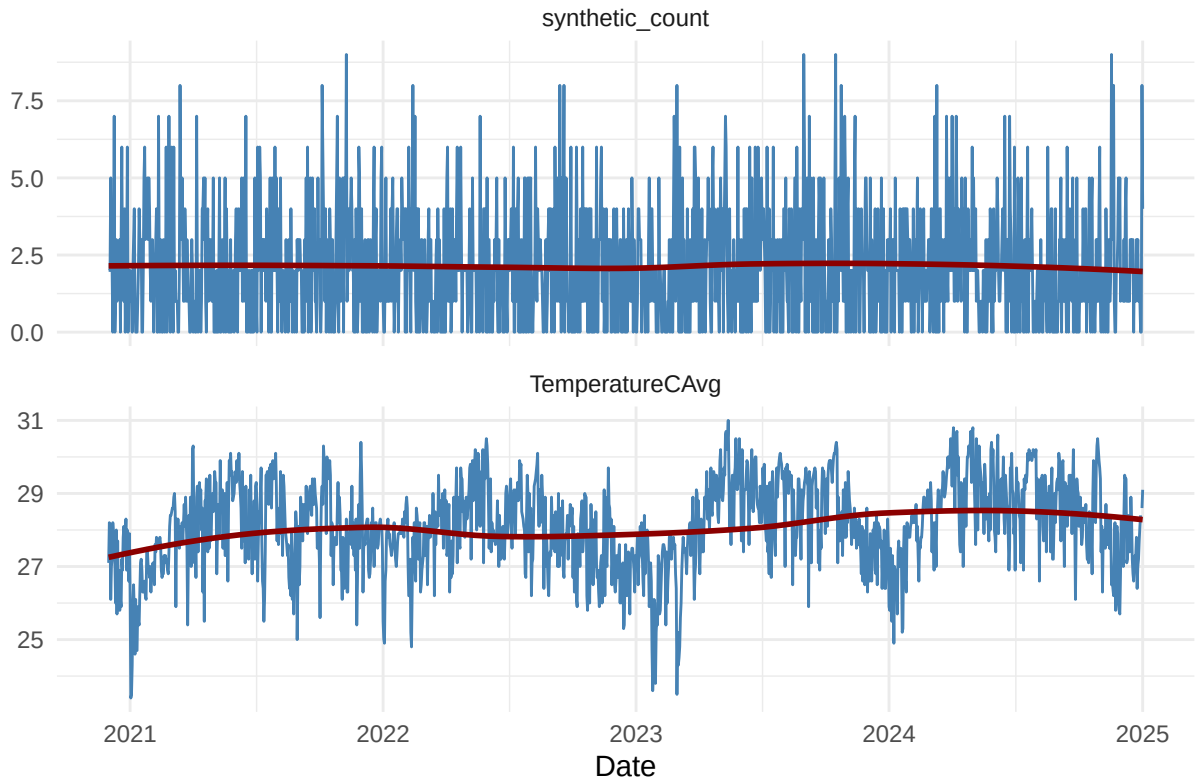
```r
changi <- read_csv("./data/changi.csv")
head(changi)
```

```
## # A tibble: 6 x 18
##   Date       TemperatureCMax TemperatureCMin TemperatureCAvg TdAvgC HrAvg
##   <date>               <dbl>           <dbl>           <dbl>  <dbl> <dbl>
## 1 2020-12-01            31.8            24.6            27.1   24.2  83.6
## 2 2020-12-02            33.5            26.1            28.2   24.7  82.7
## 3 2020-12-03            31.6            24.3            28     24.7  82.8
## 4 2020-12-04            32.9            24.4            26.1   23.6  87.2
## 5 2020-12-05            31              25.4            27     23.5  82
## 6 2020-12-06            31.3            25.5            27.2   23.2  79.2
## # i 12 more variables: WindkmhDir <chr>, WindkmhInt <dbl>, PresslevHp <dbl>,
## #   Precmm <chr>, TotClOct <dbl>, lowClOct <dbl>, SunD1h <chr>, VisKm <dbl>,
## #   station_ID <dbl>, WindkmhGust <chr>, PreselevHp <lgl>, SnowDepcm <chr>
```

```r
# loading ED visit data

ed <- read_csv("./data/Synthetic_Sampled_Temporal_Data.csv")

temporal_df <- changi %>% left_join(ed, by=c("Date"="date"))

head(temporal_df %>% select(Date, TemperatureCAvg, synthetic_count))
```

```
## # A tibble: 6 x 3
##   Date       TemperatureCAvg synthetic_count
##   <date>               <dbl>           <dbl>
## 1 2020-12-01            27.1               2
## 2 2020-12-02            28.2               2
## 3 2020-12-03            28                 2
## 4 2020-12-04            26.1               5
## 5 2020-12-05            27                 3
## 6 2020-12-06            27.2               2
```

```r
# Visualising temperature and ED visits

plotdat <- temporal_df %>%
  select(Date, TemperatureCAvg, synthetic_count) %>%
  group_by(Date) %>%
  gather(variable, value, -Date)

ggplot(plotdat, aes(x = Date, y = value)) +
  geom_line(colour = "steelblue") +
  geom_smooth(method = "loess", se=FALSE, color = "darkred") +
  facet_wrap(~variable, nrow=2, scales="free_y") +
  labs(title = "", x = "Date", y = "") +
  theme_minimal()
```



3. Exploring Lag Effects

Lag refers to how previous observations influence current outcomes. In time-series modeling, it's common to create lagged variables to account for delayed effects—for example, yesterday's weather influencing today's ED visits.

We will now create lagged versions of both the ED visits and temperature variables.

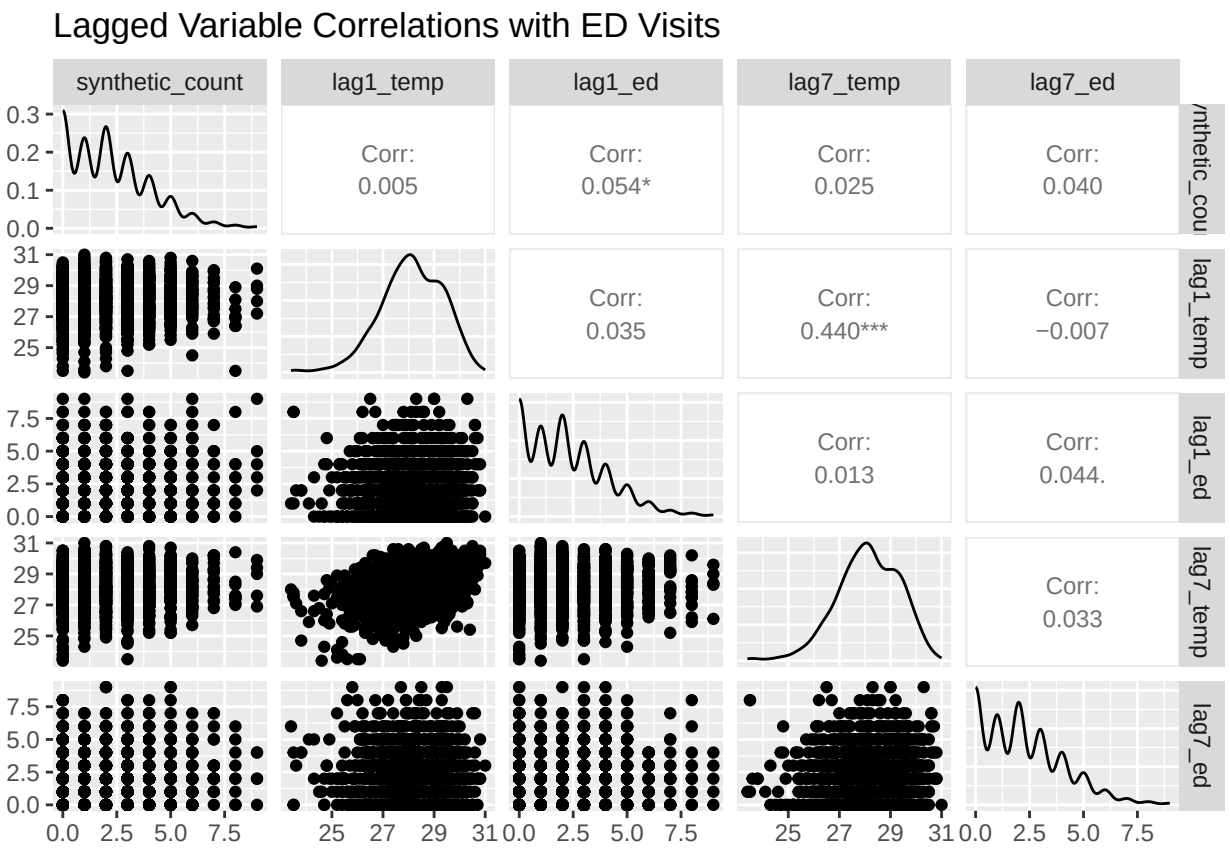
```
temporal_df <- temporal_df %>%
  arrange(Date) %>%
  mutate(
    lag1_temp = lag(TemperatureCAvg, 1),
    lag1_ed = lag(synthetic_count, 1),
    lag7_temp = lag(TemperatureCAvg, 7),
    lag7_ed = lag(synthetic_count, 7)
  )

head(temporal_df %>% select(Date, TemperatureCAvg, lag1_temp, synthetic_count, lag1_ed))
```

```
## # A tibble: 6 x 5
##   Date       TemperatureCAvg lag1_temp synthetic_count lag1_ed
##   <date>               <dbl>     <dbl>           <dbl>   <dbl>
## 1 2020-12-01            27.1      NA                 2      NA
## 2 2020-12-02            28.2      27.1               2       2
## 3 2020-12-03            28        28.2               2       2
## 4 2020-12-04            26.1      28                 5       2
```

```
## 5 2020-12-05            27        26.1              3        5
## 6 2020-12-06            27.2      27                2        3
```

Here we plot lagged temperature and lagged ED counts against the current ED count to access possible correlations.

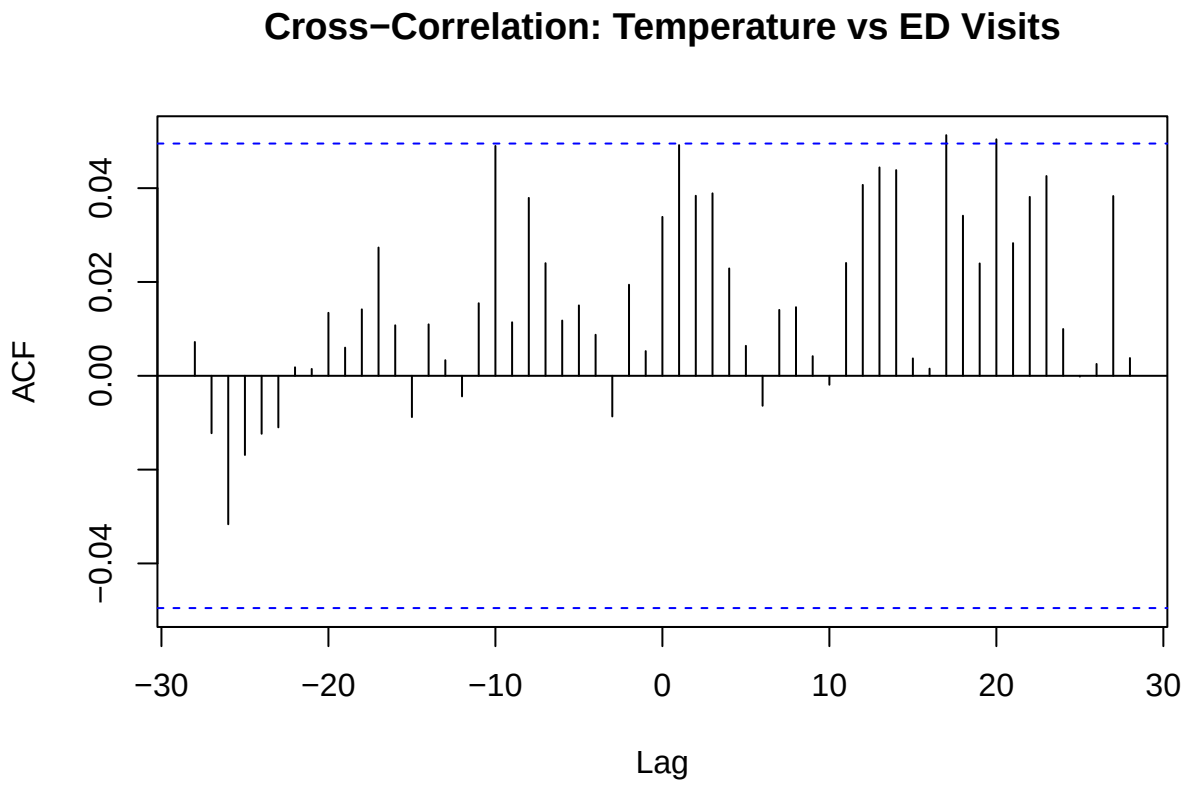
```
library(GGally)

lag_plot_data <- temporal_df %>%
  select(synthetic_count, lag1_temp, lag1_ed, lag7_temp, lag7_ed) %>%
  drop_na()

ggpairs(lag_plot_data,
        upper = list(continuous = wrap("cor", size = 3)),
        title = "Lagged Variable Correlations with ED Visits")
```



Cross-correlation plots can be useful for detemining at which lag the relationship between two series is strongest.

```
# Drop rows where either variable is NA before computing cross-correlation
ccf_data <- temporal_df %>%
  select(Date, TemperatureCAvg, synthetic_count) %>%
  drop_na()

ccf(
  x = ccf_data$TemperatureCAvg,
  y = ccf_data$synthetic_count,
  main = "Cross-Correlation: Temperature vs ED Visits"
```

```
)
```



The cross-correlation function (CCF) between daily average temperature and synthetic ED visits revealed statistically significant positive correlations at lags between 10 and 25 days. This suggests that increases in temperature may be associated with higher ED utilization approximately 1 to 3 weeks later. These findings warrant further exploration using distributed lag models.