ETS Model

Yifan Han

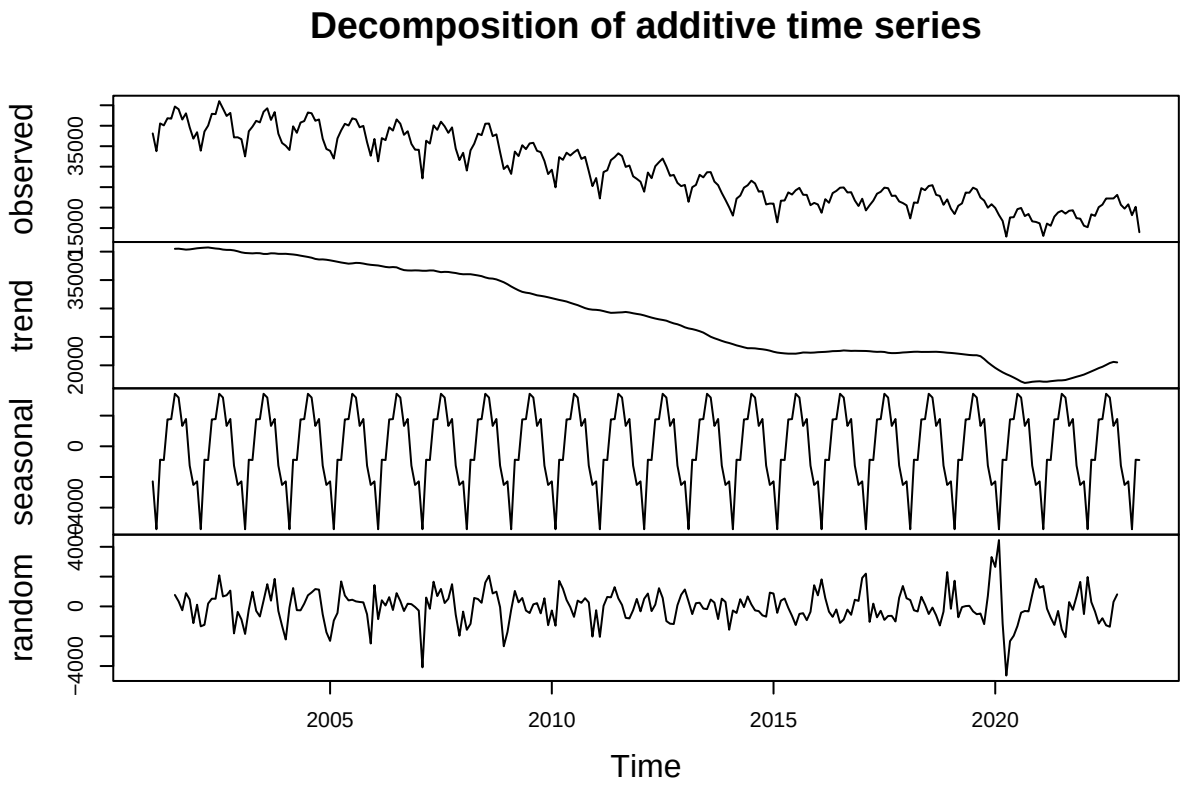
2024-05-23

Data Preparation

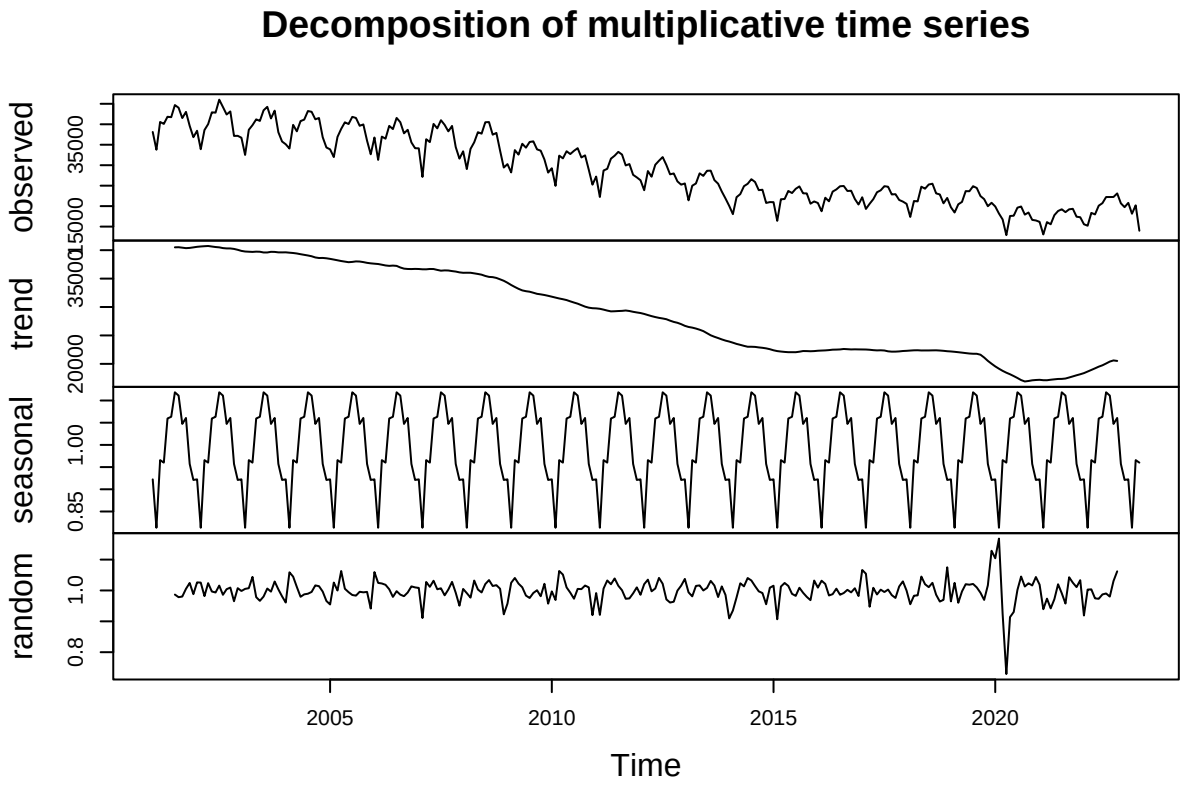
```
# Load data
data <- read.csv('/Users/yifan_/Desktop/Classes/Time series/final/crime_data.csv')

# Convert to time series data
ts_data <- ts(data[, c("total_cases")], start=c(2001, 1), frequency=12)

# Define training and test sets
train_end <- c(2021, 12)
test_start <- c(2022, 1)
test_end <- c(2022, 12)
ts_train <- window(ts_data, end=train_end)
ts_test <- window(ts_data, start=test_start, end=test_end)
```

Seasonal decomposition

```
# Perform seasonal decomposition with an additive model
decomp_additive <- decompose(ts_data, type="additive")

# Perform seasonal decomposition with a multiplicative model
decomp_multiplicative <- decompose(ts_data, type="multiplicative")

# Plot the decompositions
par(mfrow=c(2, 1))
plot(decomp_additive)
```




```
plot(decomp_multiplicative)
```




```
# Determine the type of seasonality
additive_std <- sd(decomp_additive$seasonal, na.rm=TRUE)
multiplicative_std <- sd(decomp_multiplicative$seasonal, na.rm=TRUE)
```

```r
seasonality_type <- ifelse(additive_std < multiplicative_std, "additive", "multiplicative")
print(paste("The seasonality type is:", seasonality_type))
```

```
## [1] "The seasonality type is: multiplicative"
```

Model

```r
# Fit ETS model with "MAM"
ets_mam <- ets(ts_train, model="MAM")
forecast_mam <- forecast(ets_mam, h=length(ts_test))

# Fit ETS model with "MMM"
ets_mmm <- ets(ts_train, model="MMM")
forecast_mmm <- forecast(ets_mmm, h=length(ts_test))

# Calculate RMSE
rmse_mam <- rmse(ts_test, forecast_mam$mean)
rmse_mmm <- rmse(ts_test, forecast_mmm$mean)
print(paste("RMSE for MAM Model:", round(rmse_mam, 2)))
```

```
## [1] "RMSE for MAM Model: 2096.47"
```

```r
print(paste("RMSE for MMM Model:", round(rmse_mmm, 2)))
```

```
## [1] "RMSE for MMM Model: 2090.74"
```

```r
# Calculate sMAPE
smape_mam <- Metrics::smape(ts_test, forecast_mam$mean)
smape_mmm <- Metrics::smape(ts_test, forecast_mmm$mean)
print(paste("SMAPE for MAM Model:", round(smape_mam, 2)))
```

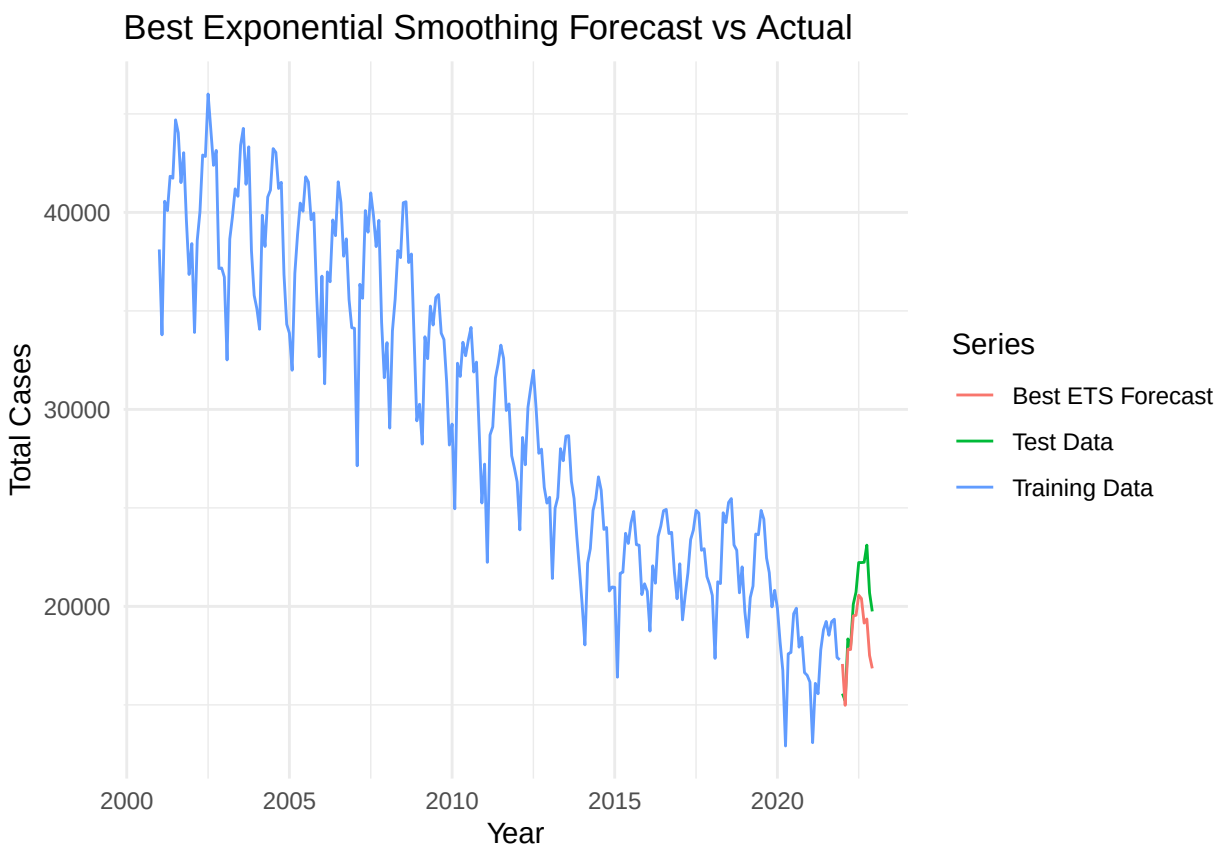
```
## [1] "SMAPE for MAM Model: 0.09"
```

```r
print(paste("SMAPE for MMM Model:", round(smape_mmm, 2)))
```

```
## [1] "SMAPE for MMM Model: 0.09"
```

Plot

```r
autoplot(ts_train, series="Training Data") +
  autolayer(ts_test, series="Test Data") +
  autolayer(forecast_mmm$mean, series="Best ETS Forecast") +
  ggtitle("Best Exponential Smoothing Forecast vs Actual") +
  xlab("Year") + ylab("Total Cases") +
  theme_minimal() +
  guides(colour=guide_legend(title="Series"))
```


Best Exponential Smoothing Forecast vs Actual
Total Cases
40000
30000
20000
2000
2005
2010
2015
2020
Year
Series
Best ETS Forecast
Test Data
Training Data