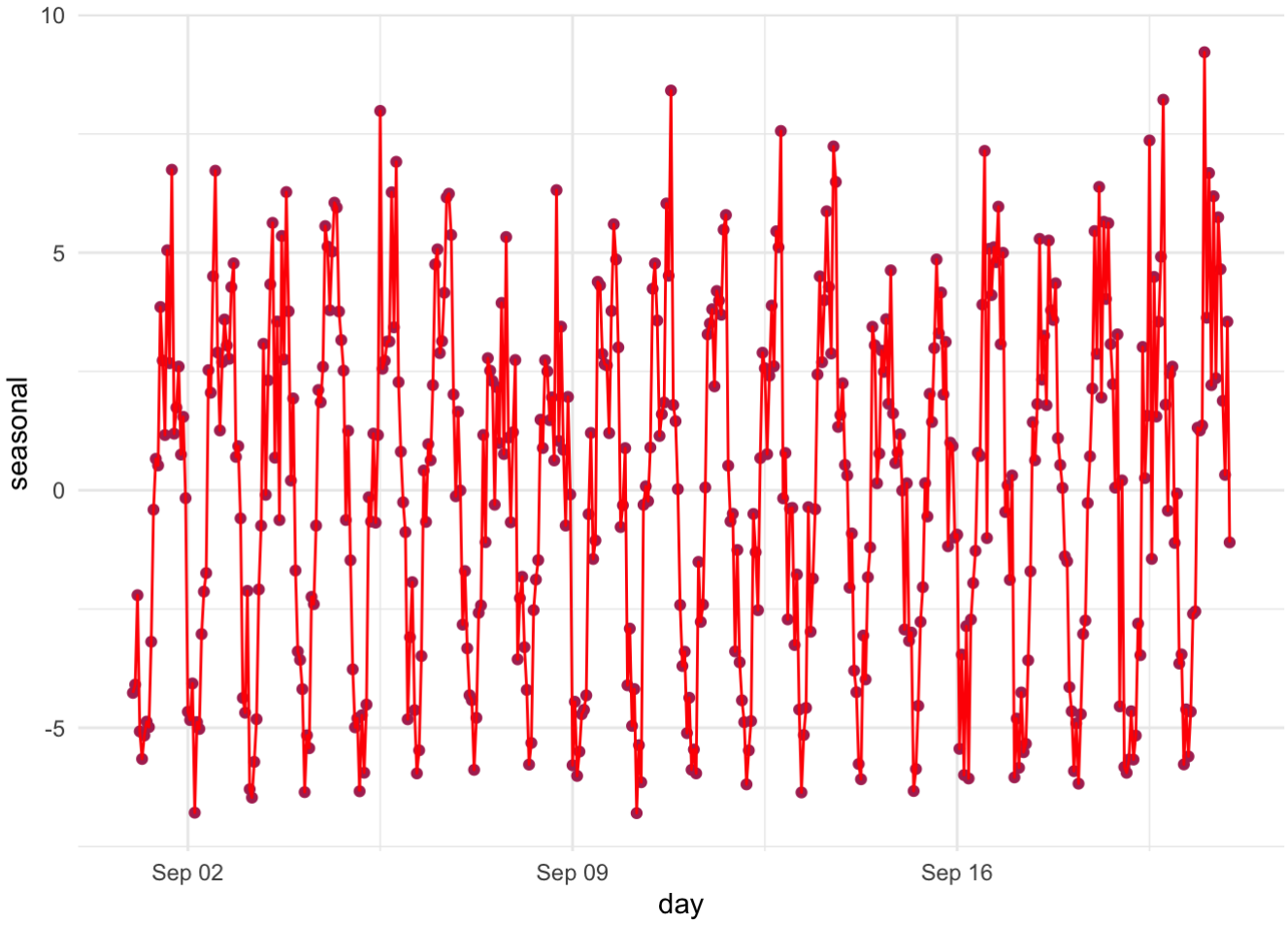
Shepherd_M5_Lab_DSCI609

Emily Shepherd

2022-09-18

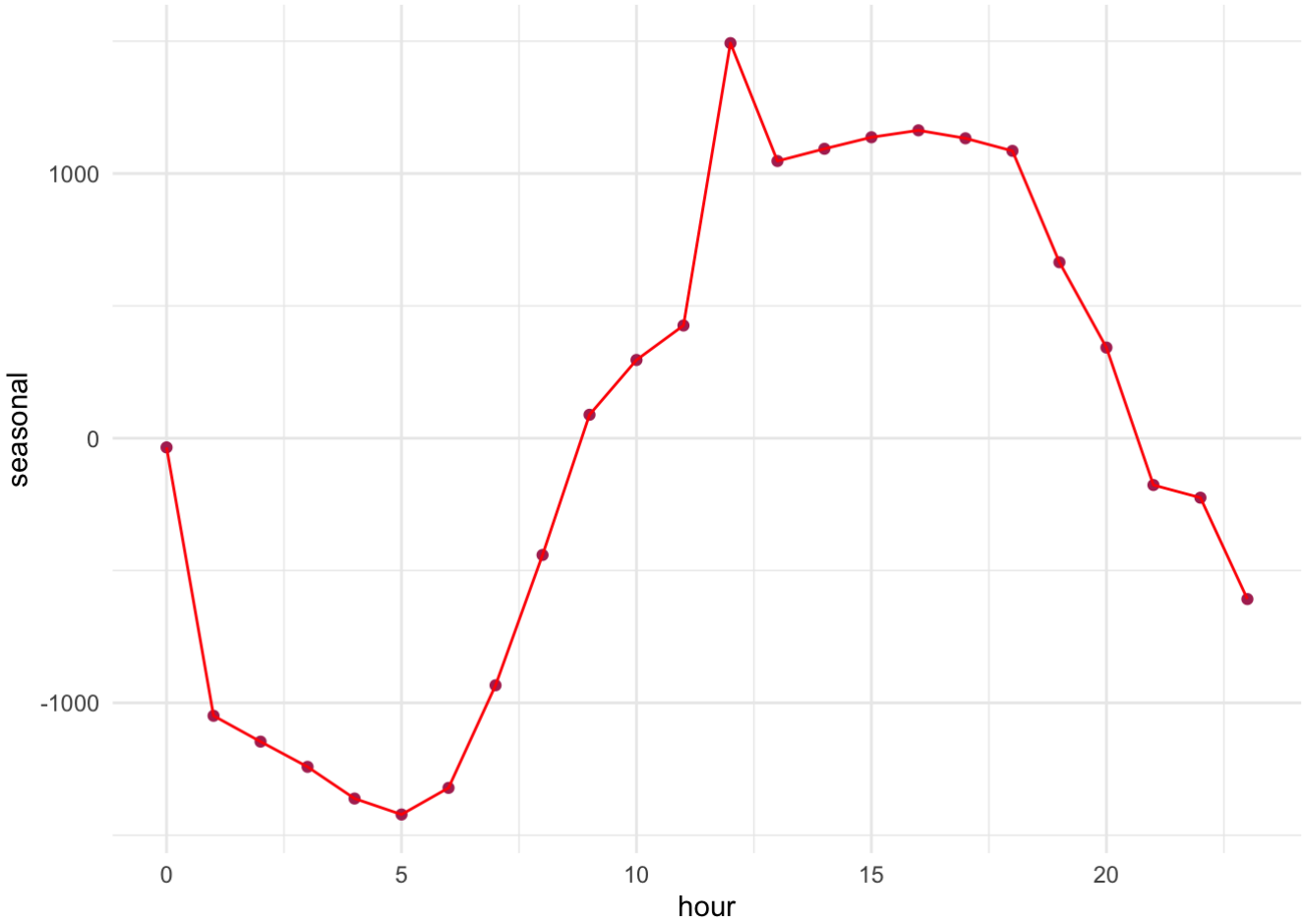
Part 1: Seasonality Analysis

```
plot1
```



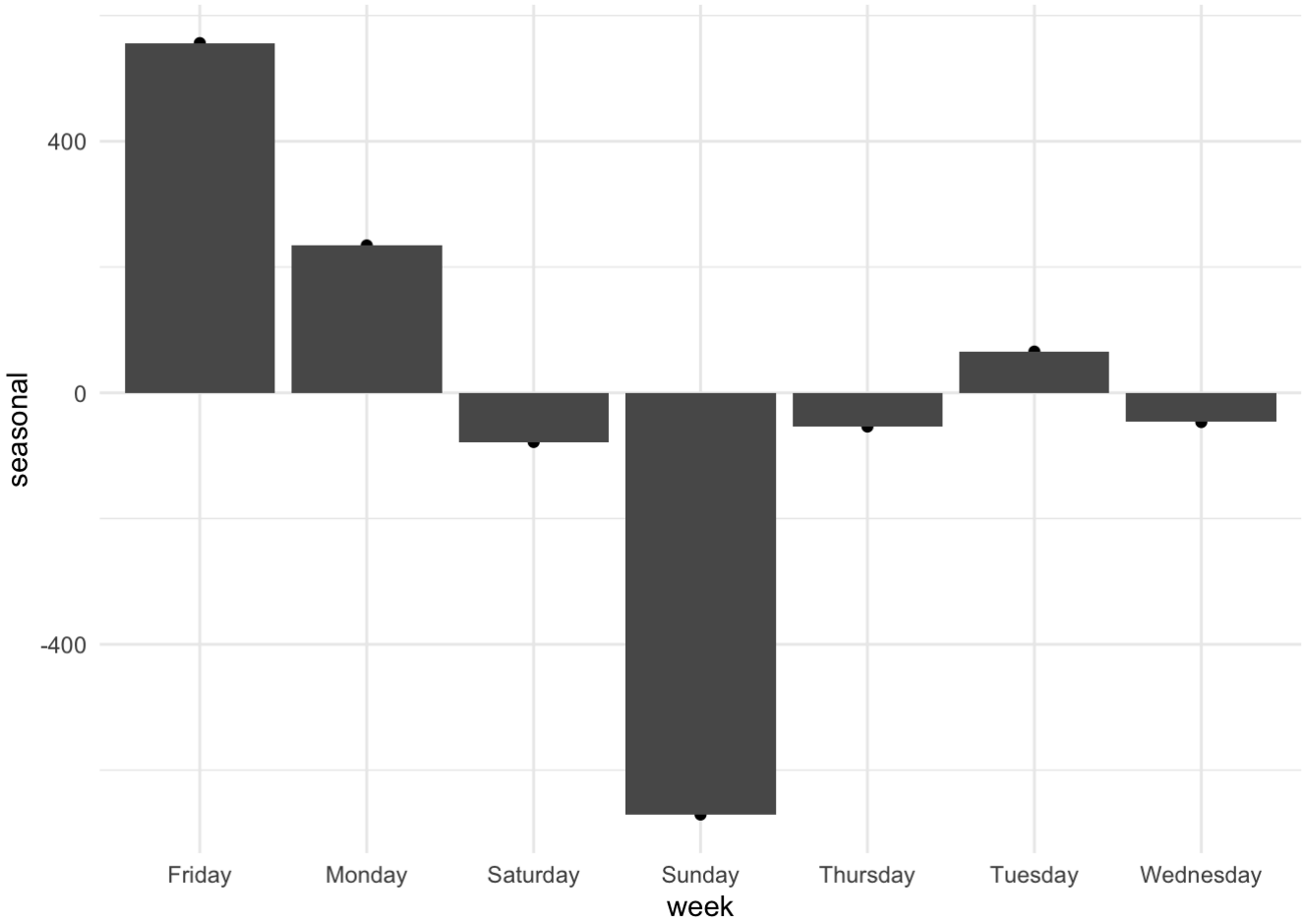
The plot above shows the number of thefts occurring in Chicago throughout the month of September 2019. From the plot we can see there appears to be trends. Because of this, the seasonality of the data needs to be examined at the hourly, weekly, and monthly levels.

```
plot2
```



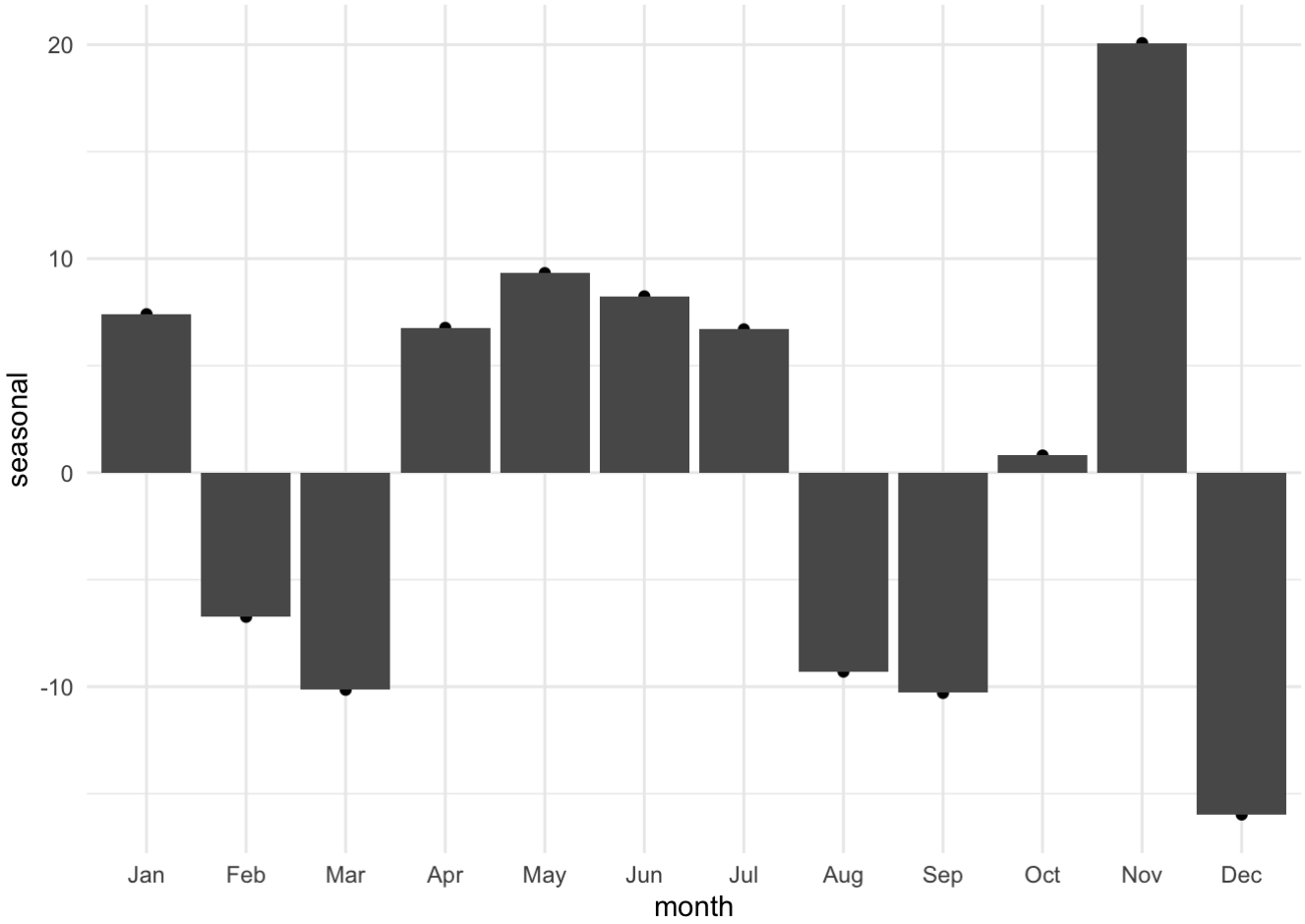
The plot above is the hourly seasonality plot. From this plot, we can see that the most thefts occur around noon. The least number of thefts occur at 5AM.

```
plot3
```



The plot above is the weekly seasonality. We can see from the plot that the greatest number of thefts occur on Friday and the fewest thefts of Sunday.

```
plot4
```

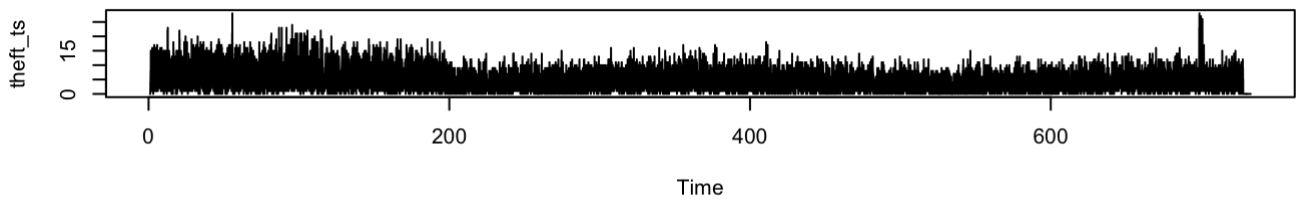


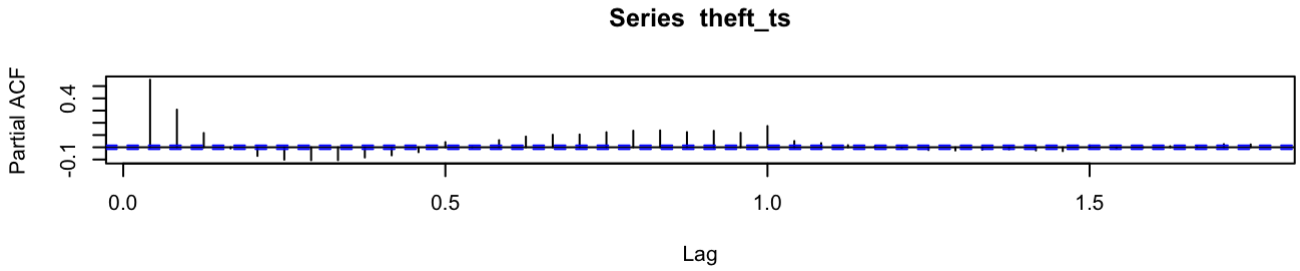
Lastly, the plot above shows the monthly seasonality. From the plot, we can see that the largest number of thefts in Chicago occur in November; while the fewest occur in December.

Part 2: Model Building and Forecasting

Before building the models, the data needs to be evaluated to see if it is stationary.

```
par(mfrow = c(3,1))
ts.plot(theft_ts)
acf(theft_ts)
pacf(theft_ts)
```





From the above graphs, it appears that our data is not stationary. To be sure, we will complete hypothesis testing.

Test 1: Dickey-Fuller Test

```
adf.test(theft_ts) #Reject H0 Time Series is Stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  theft_ts
## Dickey-Fuller = -9.8114, Lag order = 25, p-value = 0.01
## alternative hypothesis: stationary
```

From this test, because our p-value is less that 0.05, we reject the null hypothesis concluding that our data is stationary. However, we will continue with the KPSS to be sure, since the graphs appear to not be stationary.

Test 2: Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

```
kpss.test(theft_ts, null = "Trend")
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  theft_ts
## KPSS Trend = 4.8234, Truncation lag parameter = 14, p-value = 0.01
```

```
kpss.test(theft_ts, null = "Level")
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  theft_ts
## KPSS Level = 28.541, Truncation lag parameter = 14, p-value = 0.01
```

For the KPSS, the null hypothesis is that the data is stationary. In both tests, the p-value is less than 0.05, so we reject the null hypothesis. Therefore, we would conclude that the data is not stationary.

Transforming Data:

We need to check to see the number of differences needed to transform the data to stationary. We will check for both the ADF and KPSS tests.

```
ndiffs(theft_ts, test = "adf")
```

```
## [1] 0
```

```
ndiffs(theft_ts, test = "kpss")
```

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

For the adf test, 0 differences are needed. This is expected because the conclusion of the adf test was that the data was stationary. For the kpss test, we would need one difference to transform the data to stationary.

```
theft_ts_stationary <- diff(theft_ts, differences = 1)
adf.test(theft_ts_stationary)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  theft_ts_stationary
## Dickey-Fuller = -59.988, Lag order = 25, p-value = 0.01
## alternative hypothesis: stationary
```

```
kpss.test(theft_ts_stationary, null = "Trend")
```

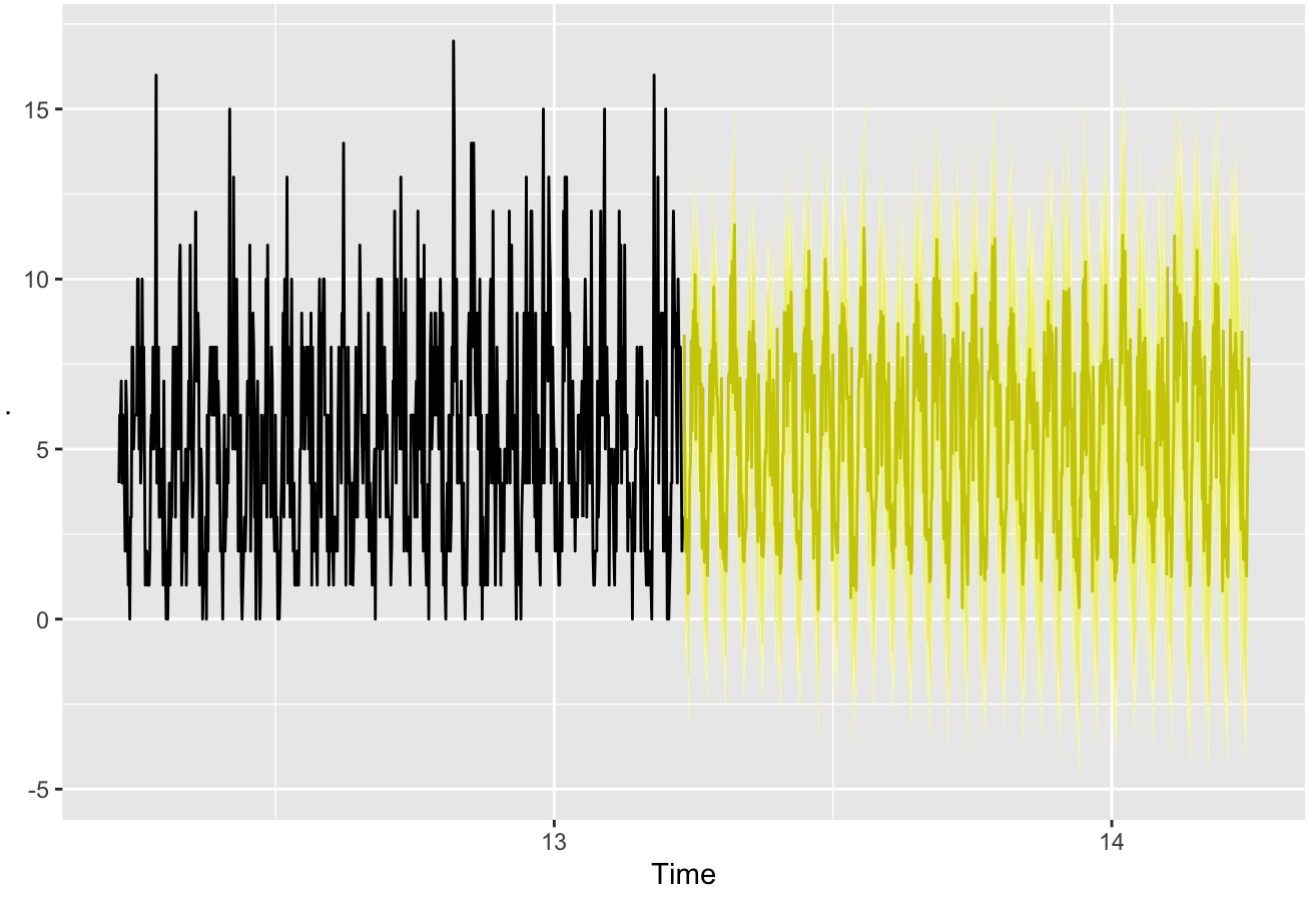
```
##
##  KPSS Test for Trend Stationarity
##
## data:  theft_ts_stationary
## KPSS Trend = 0.0006048, Truncation lag parameter = 14, p-value = 0.1
```

After transforming the data using one difference, both the adf test and kpss test indicate the data is stationary. It is now ready to fit a model.

Creating a Model:

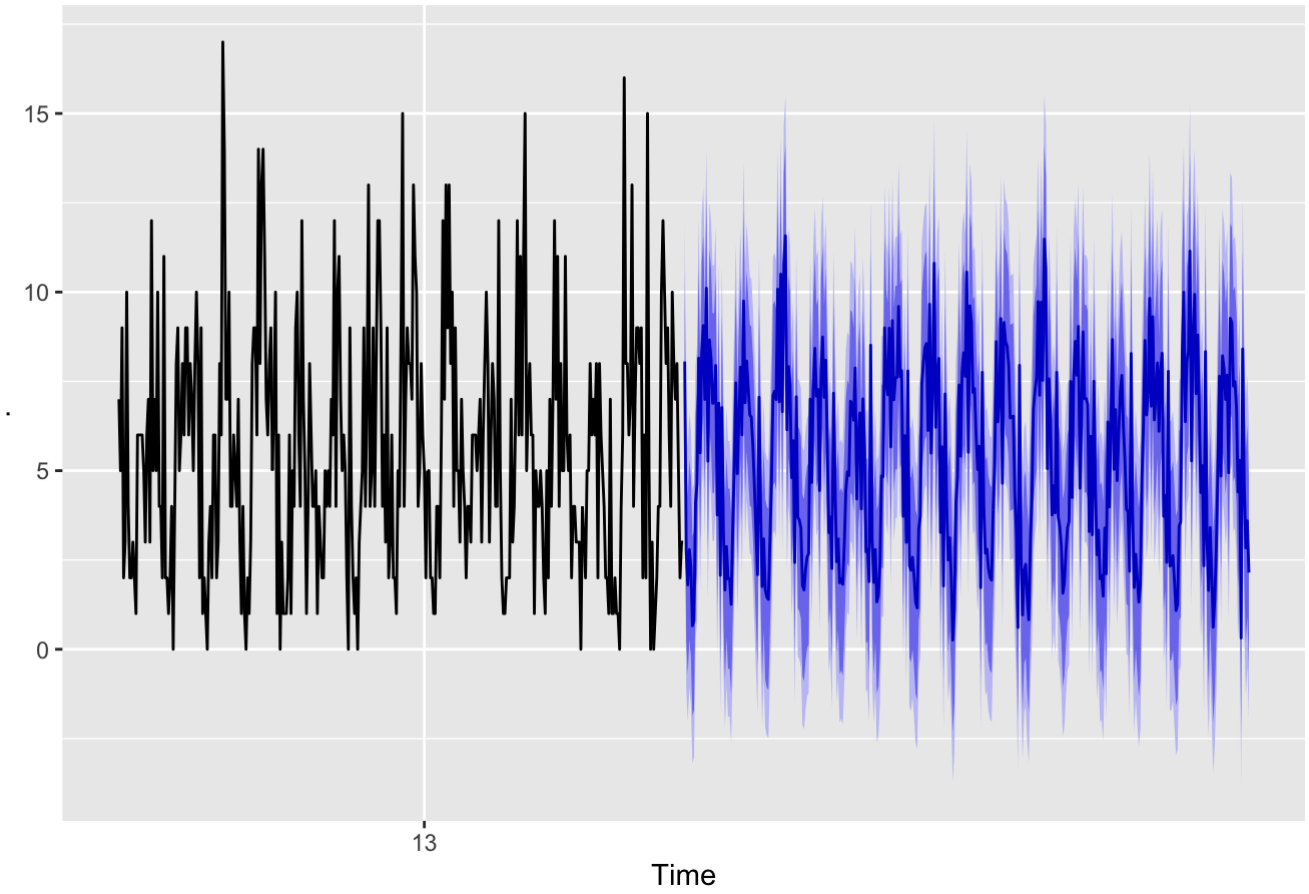
Holt-Winter:

```
theft_hw <-stlm(theft_multi, method = "ets")
forecast_hw <- forecast(theft_hw, h = 730)
theft_multi %>%
  tail(730) %>%
  autoplot() +
  autolayer(forecast_hw, series = "Multi-Seasonal Holt Winter", color = "yellow")
```



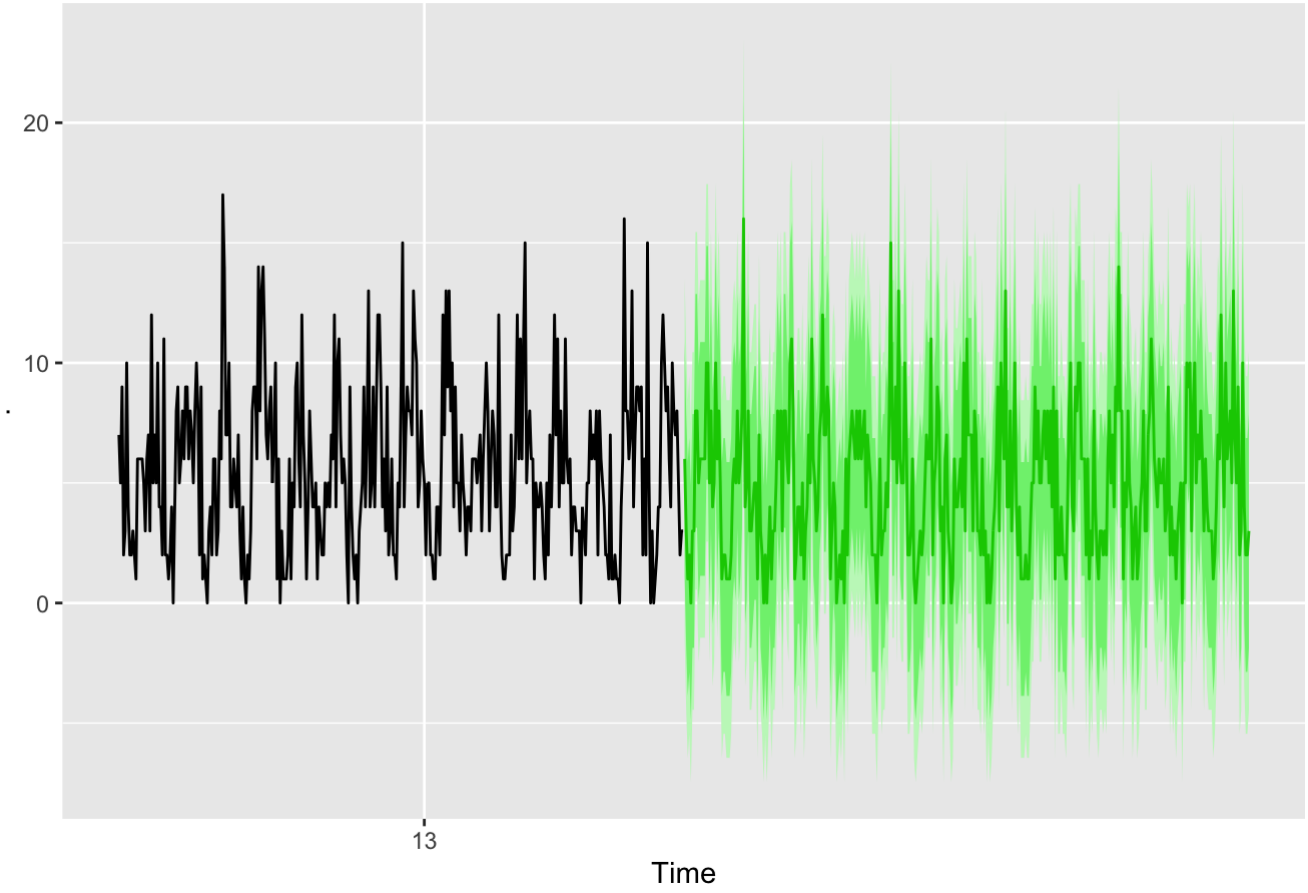
ARIMA:

```
theft_arima <-stlm(theft_multi, method = "arima")
forecast_arima <- forecast(theft_arima, h = 365)
theft_multi %>%
  tail(365) %>%
  autoplot() +
  autolayer(forecast_arima, series = "Multi-Seasonal ARIMA", color = "blue")
```



Seasonal Naive:

```
theft_snaive = snaive(y = theft_multi, h = 365)
forecast_snaive <- forecast(object = theft_snaive)
theft_multi %>%
  tail(365) %>%
  autoplot() +
  autolayer(forecast_snaive, series = "Seasonal Naive Method", color = "green")
```



Analysis to Determine Which Model is Best:

Holt-Winter:

```
accuracy(forecast_hw, head(theft_test$Theft,365))
```

```
##                       ME     RMSE      MAE  MPE MAPE      MASE        ACF1
## Training set -0.008094165 1.942284 1.480617  NaN  Inf 0.5312174 0.02914408
## Test set     -0.181185519 2.819761 2.128717 -Inf  Inf 0.7637436          NA
```

ARIMA:

```
accuracy(forecast_arima, head(theft_test$Theft,365))
```

```
##                       ME     RMSE      MAE  MPE MAPE      MASE         ACF1
## Training set -0.009461599 1.933510 1.474164  NaN  Inf 0.5289023 0.003834374
## Test set     -0.161648518 2.818008 2.128809 -Inf  Inf 0.7637764           NA
```

Seasonal Naive:

```
accuracy(forecast_snaive, head(theft_test$Theft,365))
```

```
##                       ME     RMSE      MAE  MPE MAPE      MASE      ACF1
## Training set -0.1998022 3.797710 2.846192 -Inf  Inf 1.0211599 0.1647482
## Test set      0.1890411 3.238044 2.424658 -Inf  Inf 0.8699213        NA
```

By comparing the RMSE of each model, we would conclude that the ARIMA model fit best, because it had the smallest RMSE.

Checking the Residuals for Normality:

```
ks.test(theft_arima$residuals, y = "pnorm", alternative = "two.sided")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  theft_arima$residuals
## D = 0.13615, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Since the p-value is less that 0.05, we would conclude that the residuals are not normally distributed.

```
Box.test(theft_arima$residuals, type = "Ljung-Box", lag = 2*365)
```

```
##
##  Box-Ljung test
##
## data:  theft_arima$residuals
## X-squared = 2923.3, df = 730, p-value < 2.2e-16
```

Since the p-value is less than 0.05, we would conclude that autocorrelation is present in the model.