Technical Notebook for Team 1 Final Project ADS 506

Team 1

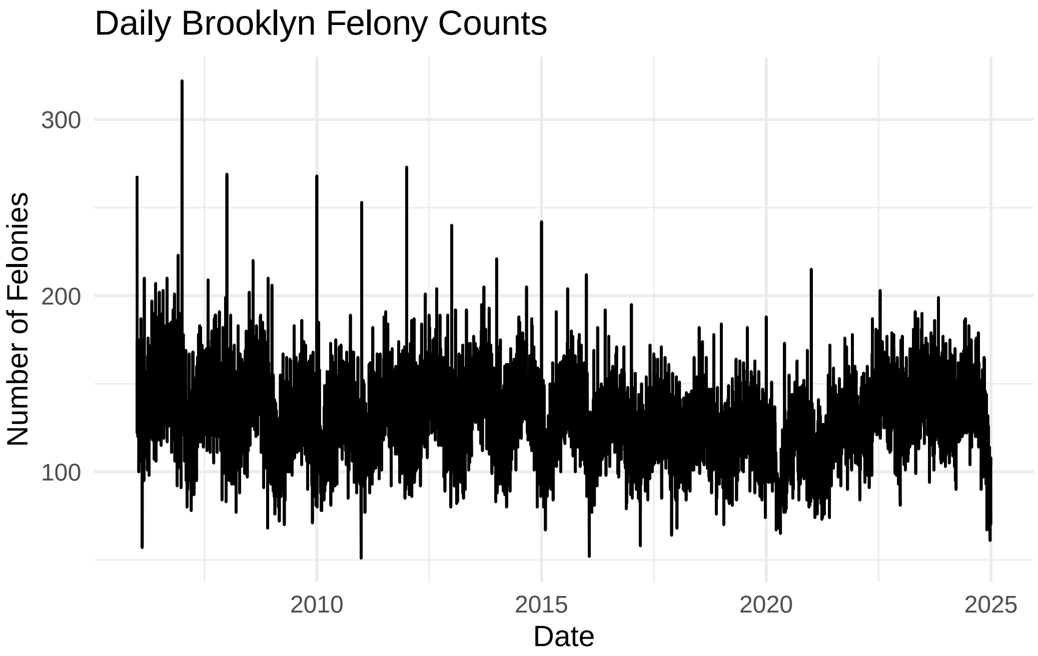
```r
library(readr)
library(hms)
library(fpp3)
library(patchwork)
library(ggtime)
library(RSocrata)
library(stringr)
library(tidyverse)
library(gt)
library(fable.prophet)
```

Exploratory Data Analysis

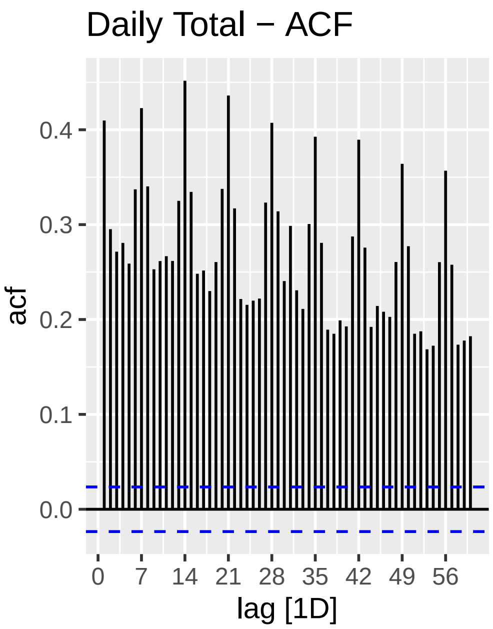
```r
# Pull Brooklyn felony records from NYPD API (server-side filter)
# There are 9.49 million records in the Historic data set, we only want
# the felonies committed in Brooklyn
# (excluding misdemeanors, violations, or any other boroughs)
base_url <- "https://data.cityofnewyork.us/resource/qgea-i56i.csv"

query_url <- paste0(
  base_url,
  "?$where=boro_nm='BROOKLYN' AND law_cat_cd='FELONY'",
  "&$select=cmplnt_fr_dt, cmplnt_fr_tm, pd_cd, pd_desc"
)

brooklyn_raw <- read.socrata(query_url) |>
  as_tibble()
```

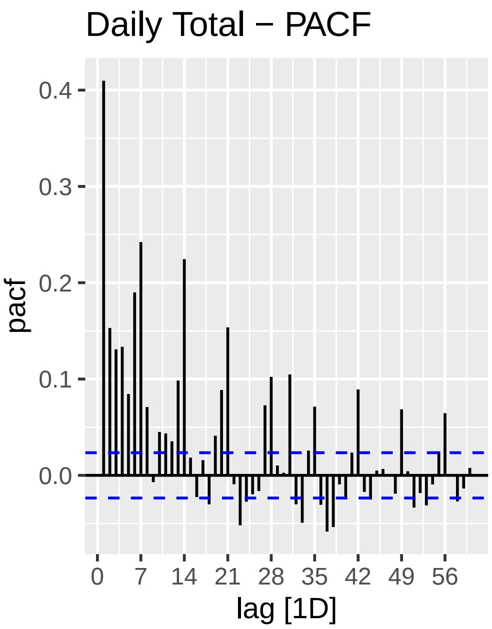
```r
# Clean date/time
brooklyn_clean <- brooklyn_raw |>
  mutate(
    # ensure class is Date
    felony_date = as_date(cmplnt_fr_dt),
    felony_time = hms(cmplnt_fr_tm),
    felony_hour = hour(felony_time)
  ) |>
  filter(
    !is.na(felony_date),
    !is.na(felony_hour),
    felony_date >= as_date("2006-01-01")
  ) |>
  select(felony_date, felony_hour, pd_cd, pd_desc)
write_csv(brooklyn_clean, "brooklyn_felonies_clean_records.csv")

# Create CSV and tsibble of the DAILY total felony counts
daily_total <- brooklyn_clean |>
  count(felony_date, name = "felony_count") |>
  arrange(felony_date)

write_csv(daily_total, "daily_total_felonies.csv")

daily_total_ts <- daily_total |>
  as_tsibble(index = felony_date)
```

```r
# View autoplot and ACF and PACF plots of daily total felonies
daily_total_ts |>
  autoplot(felony_count) +
  labs(
    title = "Daily Brooklyn Felony Counts",
    x = "Date",
    y = "Number of Felonies"
  ) +
  theme_minimal()
```

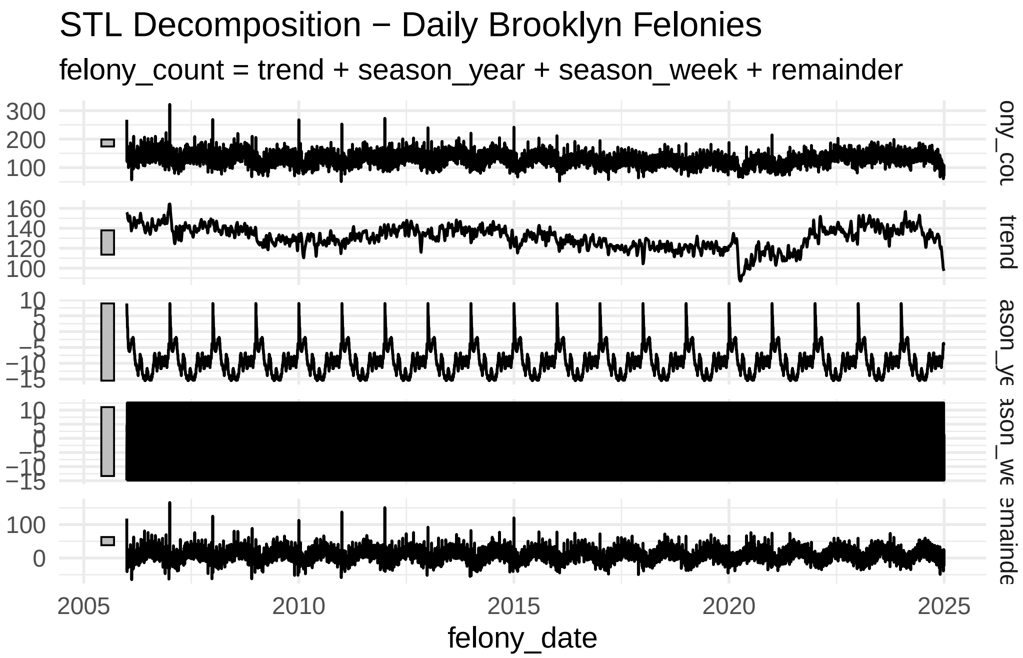


```
daily_acf <- daily_total_ts |>
  ACF(felony_count, lag_max = 60) |>
  autoplot() +
  ggtitle("Daily Total - ACF")

daily_pacf <- daily_total_ts |>
  PACF(felony_count, lag_max = 60) |>
  autoplot() +
  ggtitle("Daily Total - PACF")

daily_acf + daily_pacf
```



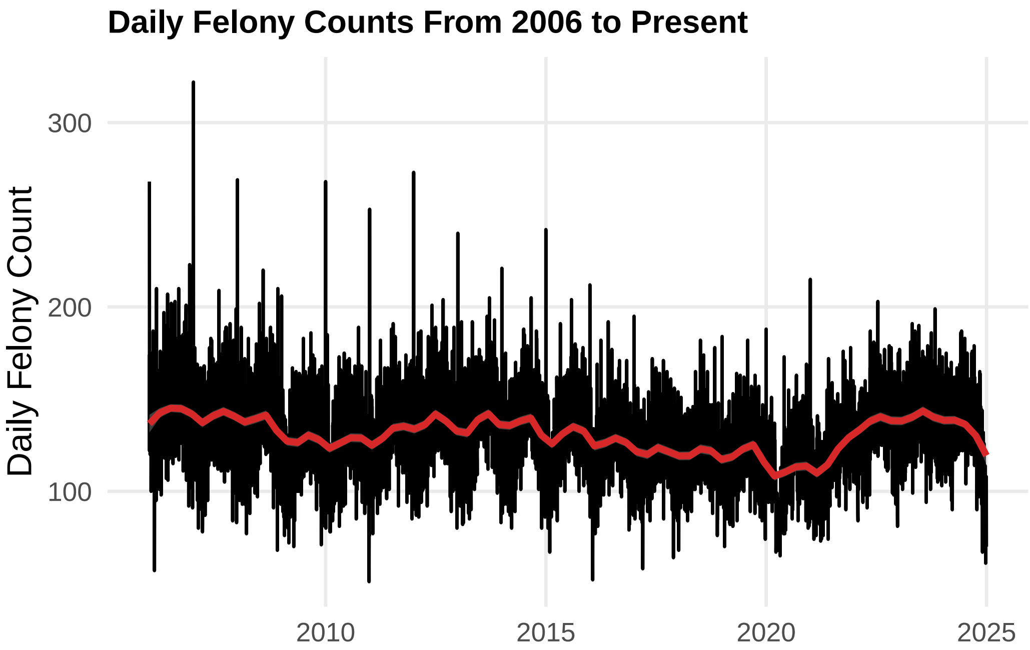


```r
# STL Decomposition
daily_total_ts |>
  model(
    STL(felony_count ~ trend(window = 21) + season(window = "periodic"))
  ) |>
  components() |>
  autoplot() +
  labs(title = "STL Decomposition - Daily Brooklyn Felonies") +
  theme_minimal()
```

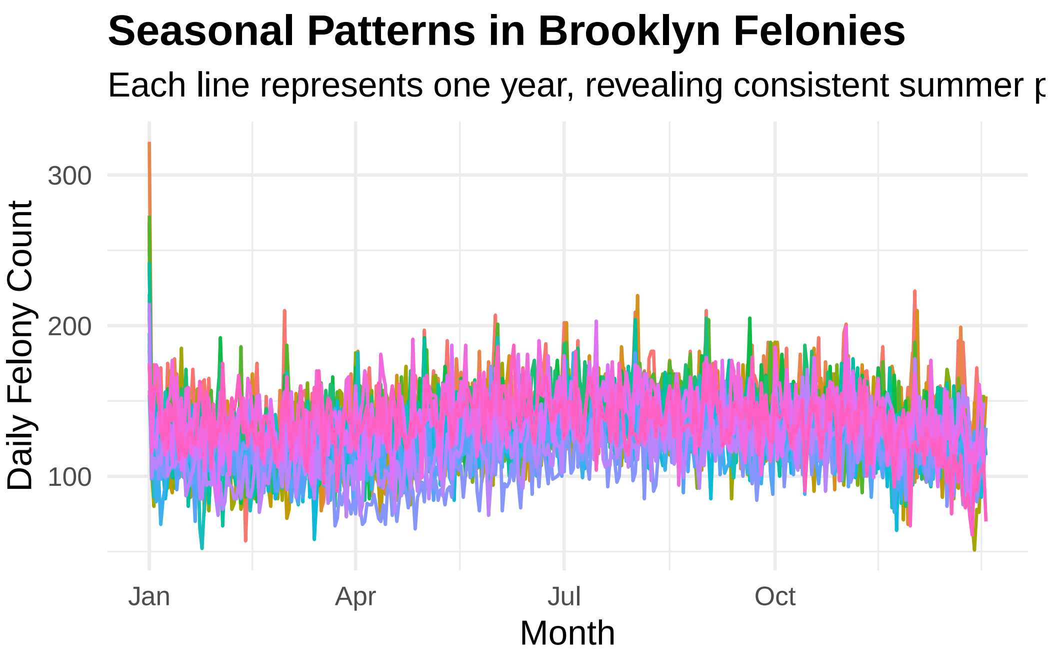


Background/setting visualizations

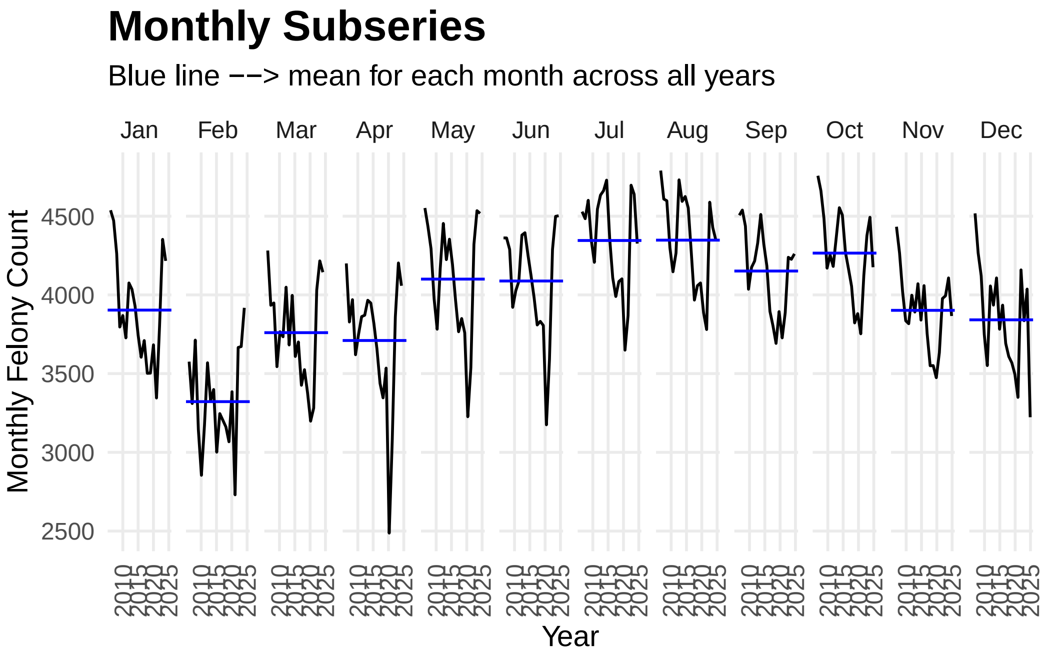
```
# Visualization 1: Overall trend of daily felony counts over time
viz1_overall_trend <- daily_total_ts |>
  autoplot(felony_count) +
  geom_smooth(aes(y = felony_count), method = "loess", span = 0.1,
              color = "#d62828", linewidth = 1.2, se = TRUE) +
  labs(
    title = "Daily Felony Counts From 2006 to Present",
    x = NULL,
    y = "Daily Felony Count"
  ) +
  theme_minimal(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold", size = 12),
    panel.grid.minor = element_blank()
  )

ggsave("01_overall_trend_autoplot.png", viz1_overall_trend, width = 12, height = 6, dpi = 30
viz1_overall_trend
```



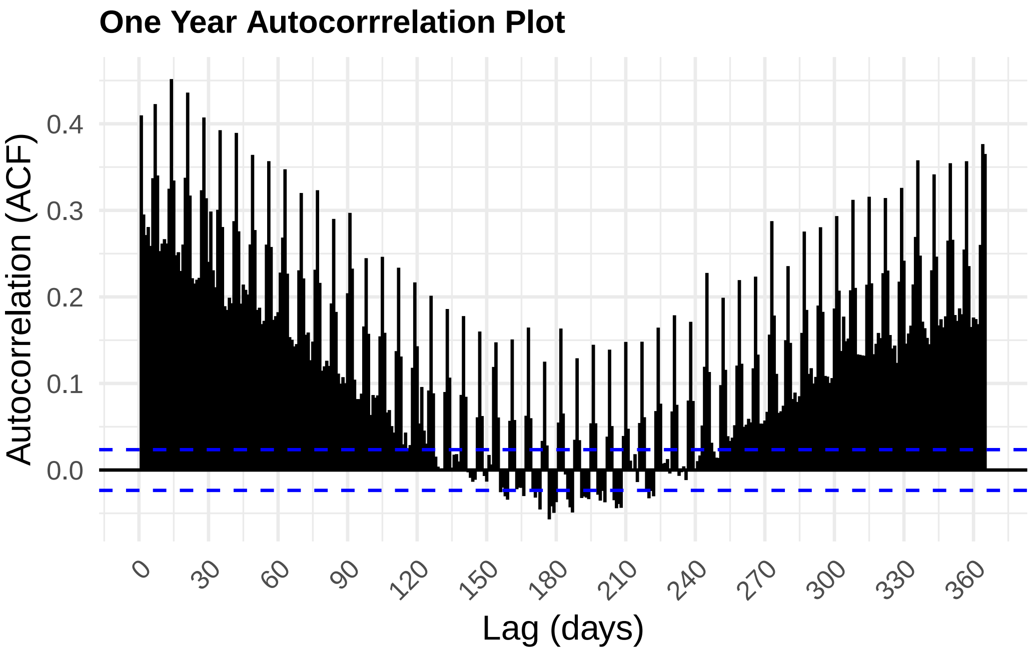
```r
# Visualization 2: Seasonal plot
viz2_seasonal <- daily_total_ts |>
  gg_season(felony_count, period = "year") +
  labs(
    title = "Seasonal Patterns in Brooklyn Felonies",
    subtitle = "Each line represents one year, revealing consistent summer peaks",
    x = "Month",
    y = "Daily Felony Count",
    color = "Year"
  ) +
  theme_minimal(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    legend.position = "none"
  )

ggsave("02_seasonal_plot.png", viz2_seasonal, width = 12, height = 6, dpi = 300)
viz2_seasonal
```



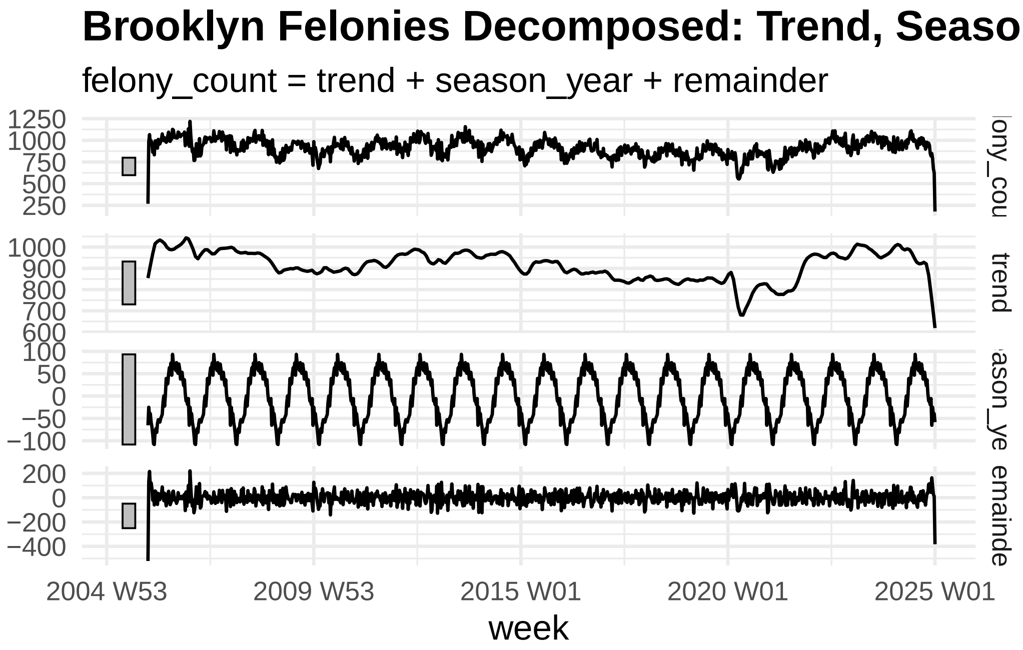
```
# Visualization 3: Subseries Plot
monthly_total_ts <- daily_total_ts |>
  index_by(year_month = ~ yearmonth(.)) |>
  summarise(felony_count = sum(felony_count))

viz3_subseries <- monthly_total_ts |>
  gg_subseries(felony_count, period = "year") +
  labs(
    title = "Monthly Subseries",
    subtitle = "Blue line --> mean for each month across all years",
    x = "Year",
    y = "Monthly Felony Count"
  ) +
  theme_minimal(base_size = 11) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)
  )

ggsave("03_subseries_plot.png", viz3_subseries, width = 12, height = 8, dpi = 300)
viz3_subseries
```



```
# Visualization 4: ACF Plot
viz4_acf <- daily_total_ts |>
  ACF(felony_count, lag_max = 365) |>
  autoplot() +
  scale_x_continuous(breaks = seq(0, 365, by = 30)) +  # Breaks every 30 days
  labs(
    title = "One Year Autocorrrelation Plot",
    x = "Lag (days)",
    y = "Autocorrelation (ACF)"
  ) +
  theme_minimal(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold", size = 12),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )

ggsave("04_acf_plot.png", viz4_acf, width = 12, height = 6, dpi = 300)
viz4_acf
```



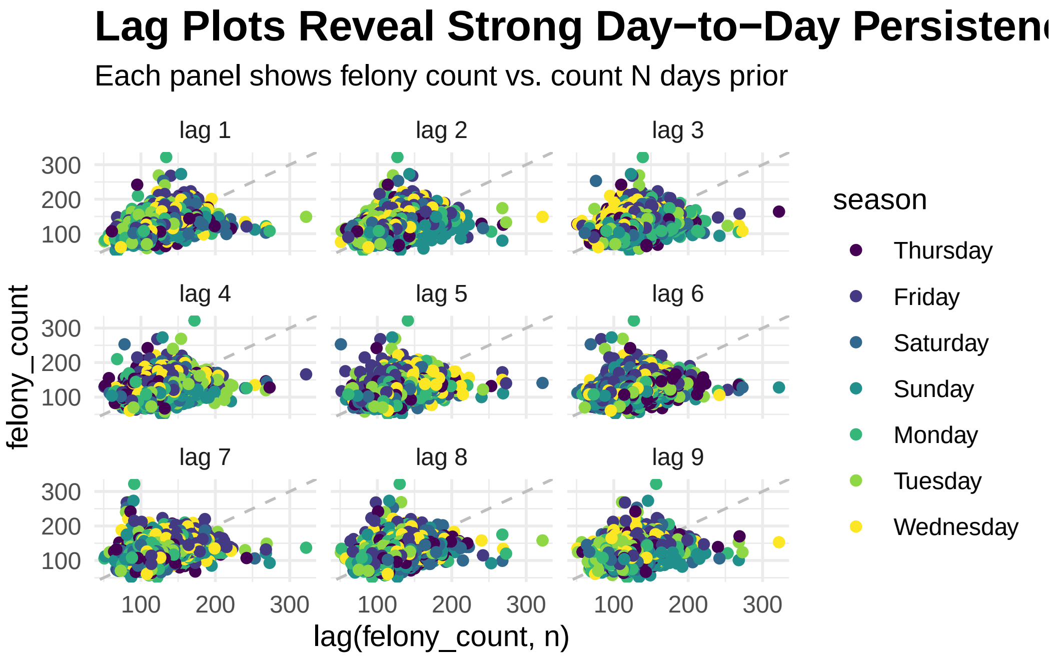
```r
# Visualization 5: STL Decomposition

# Aggregate to weekly for cleaner decomposition
weekly_total_ts <- daily_total_ts |>
  index_by(week = ~ yearweek(.)) |>
  summarise(felony_count = sum(felony_count))

viz5_stl <- weekly_total_ts |>
  model(
    STL(felony_count ~ trend(window = 21) + season(window = "periodic"))
  ) |>
  components() |>
  autoplot() +
  labs(
    title = "Brooklyn Felonies Decomposed: Trend, Seasonality, and Noise"
  ) +
  theme_minimal(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold", size = 16)
  )

ggsave("05_stl_decomposition.png", viz5_stl, width = 12, height = 8, dpi = 300)
viz5_stl
```



Brooklyn Felonies Decomposed: Trend, Season

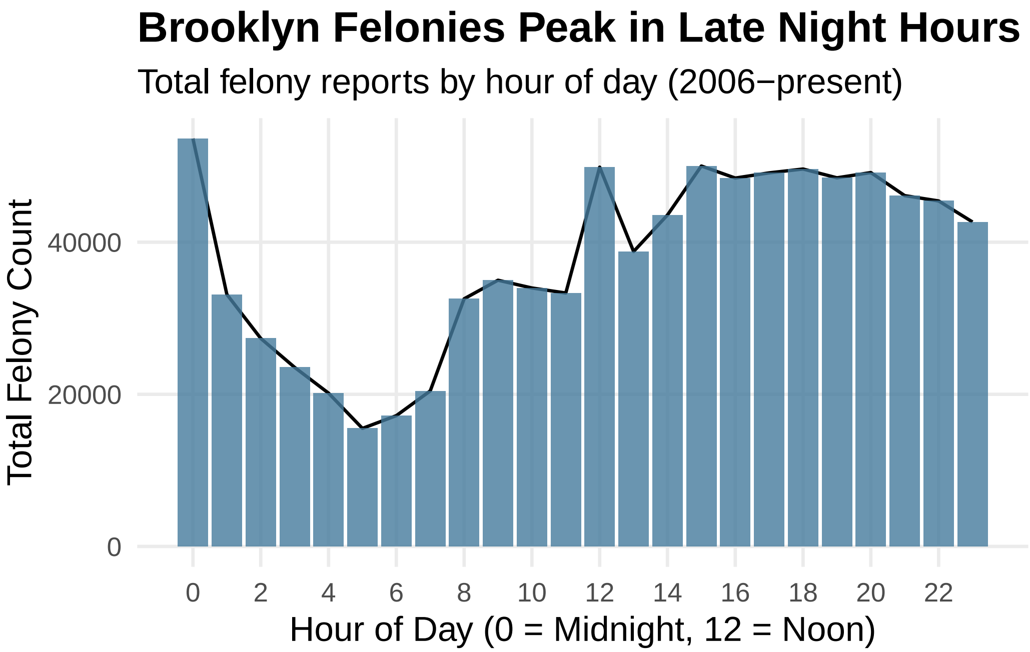
felony_count = trend + season_year + remainder

```
# Visualization 6: Lag Plot
viz6_lag <- daily_total_ts |>
  gg_lag(felony_count, lags = 1:9, geom = "point") +
  labs(
    title = "Lag Plots Reveal Strong Day-to-Day Persistence",
    subtitle = "Each panel shows felony count vs. count N days prior"
  ) +
  theme_minimal(base_size = 11) +
  theme(
    plot.title = element_text(face = "bold", size = 16)
  )

ggsave("06_lag_plots.png", viz6_lag, width = 12, height = 8, dpi = 300)
viz6_lag
```



```
# Visualization 7: Time of day analysis (Hourly patterns)

# Create a "time of day" tsibble aggregated across all dates
hourly_pattern_ts <- brooklyn_clean |>
  count(felony_hour, name = "total_count") |>
  mutate(hour = as.integer(felony_hour)) |>
  as_tsibble(index = hour)

viz7_hourly <- hourly_pattern_ts |>
  autoplot(total_count) +
  geom_col(aes(y = total_count), fill = "#457b9d", alpha = 0.8) +
  scale_x_continuous(breaks = seq(0, 23, 2)) +
  labs(
    title = "Brooklyn Felonies Peak in Late Night Hours",
    subtitle = "Total felony reports by hour of day (2006-present)",
    x = "Hour of Day (0 = Midnight, 12 = Noon)",
    y = "Total Felony Count"
  ) +
  theme_minimal(base_size = 13) +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    panel.grid.minor = element_blank()
  )
```

```
ggsave("07_hourly_patterns.png", viz7_hourly, width = 12, height = 6, dpi = 300)
viz7_hourly
```



```
# Summary statistics
summary_stats <- list(
  total_felonies = nrow(brooklyn_clean),
  date_range = paste(min(brooklyn_clean$felony_date), "to", max(brooklyn_clean$felony_date))
  years_covered = as.numeric(max(brooklyn_clean$felony_date) - min(brooklyn_clean$felony_dat
  avg_daily = mean(daily_total_ts$felony_count)
)

# Print summary
cat("\nBrooklyn Felony Data Summary\n")
```

```
Brooklyn Felony Data Summary
```

```
cat("Total Felonies:", scales::comma(summary_stats$total_felonies), "\n")
```

```
Total Felonies: 907,047
```

```r
cat("Date Range:", summary_stats$date_range, "\n")
```

```
Date Range: 2006-01-01 to 2024-12-31
```

```r
cat("Years Covered:", round(summary_stats$years_covered, 1), "\n")
```

```
Years Covered: 19
```

```r
cat("Average Daily Count:", round(summary_stats$avg_daily, 1), "\n")
```

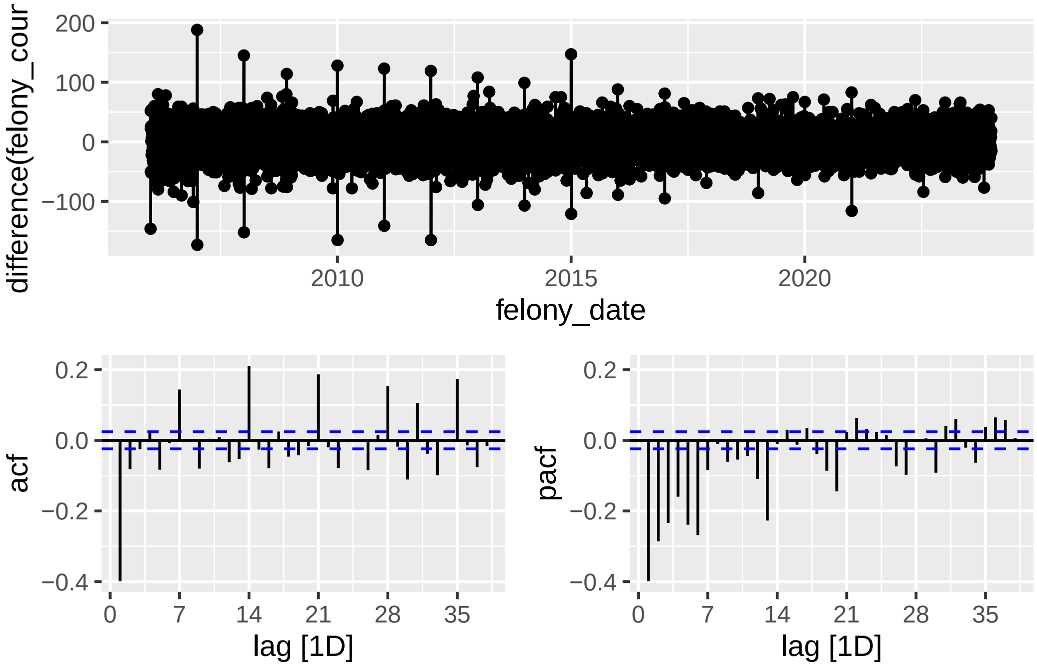
```
Average Daily Count: 130.7
```

Modeling

```r
# Create training/validation sets for the model
train <- daily_total_ts |>
  filter_index("2006-01-01" ~ "2023-12-30")
validation <- daily_total_ts |>
  filter_index("2024-01-01" ~ "2024-12-31")

# Check stationarity
train |>
  features(felony_count, c(unitroot_kpss, unitroot_ndiffs, unitroot_nsdiffs))
```

```
# A tibble: 1 x 4
  kpss_stat kpss_pvalue ndiffs nsdiffs
      <dbl>       <dbl>  <int>   <int>
1      7.23        0.01      1       0
```

```r
# Plot differenced series and ACF/PACF plots
train |>
 gg_tsdisplay(difference(felony_count), plot_type = "partial")
```



```r
# Build models
models <- train |>
 model(
   ets = ETS(felony_count),
   tslm = TSLM(felony_count ~ trend() + season()),
   arima_auto = ARIMA(felony_count),
   ma1 = ARIMA(felony_count ~ pdq(0, 1, 1) + PDQ(0, 0, 1, period = 7)),
   ar1 = ARIMA(felony_count ~ pdq(1, 1, 1) + PDQ(0, 0, 1, period = 7)),
   snaive = SNAIVE(felony_count),
   ensemble = (
     ARIMA((felony_count)) +
     ETS((felony_count)) +
     TSLM((felony_count) ~ trend() + season())) / 3,
   prophet = prophet(felony_count),
   nnetar = NNETAR(felony_count)
 )

models |> t()
```

```
           [,1]
ets        ETS(A,N,A)
tslm       TSLM
arima_auto ARIMA(2,1,2)(0,0,2)[7]
```

```
ma1         ARIMA(0,1,1)(0,0,1)[7]
ar1         ARIMA(1,1,1)(0,0,1)[7]
snaive      SNAIVE
ensemble    COMBINATION
prophet     prophet
nnetar      NNAR(37,1,19)[7]
```

Compare Models (training)

```r
models |>
  accuracy() |>
  select(.model, RMSE, MAE, MAPE) |>
  arrange(RMSE) |>
  knitr::kable(digits = 1)
```

.model	RMSE	MAE	MAPE
nnetar	12.7	9.5	7.5
ets	17.0	12.6	9.9
prophet	17.3	12.9	10.2
ensemble	17.4	12.9	10.3
arima_auto	18.3	13.8	11.0
ar1	18.6	14.1	11.2
ma1	18.8	14.2	11.3
tslm	20.5	15.7	12.7
snaive	23.9	17.7	13.9

Forecast

```r
# Forecast the validation period
h_val <- nrow(validation)
models_fc <- models |> forecast(h = h_val)

# Review accuracy to select the best model
models_fc |>
  accuracy(validation) |>
  select(.model, RMSE, MAE, MAPE) |>
  arrange(RMSE) |>
  knitr::kable(digits = 1)
```

.model	RMSE	MAE	MAPE
ets	19.4	15.0	11.4
ar1	19.9	15.3	11.8
arima_auto	19.9	15.3	11.8
ensemble	19.9	15.5	11.7
ma1	20.1	15.5	11.9
nnetar	20.3	15.5	12.3
tslm	22.3	18.1	13.3
prophet	24.9	19.6	16.2
snaive	27.7	22.7	16.9