# Forecasting Project: Projecting Total Legal Wagers on Super Bowl 59

December 2024

## Introduction

Sports betting has experienced rapid growth in the United States over the past several years, driven by the legalization of sports betting across many states and the increased popularity of mobile betting platforms. One of the most significant annual events in this industry is the Super Bowl, which consistently attracts the highest single-event betting handle of the year.

The goal of this project is to forecast the **total monthly sports betting handle** across all legal states and platforms, with a particular focus on predicting how much Americans will wager on the next Super Bowl. To achieve this, I will analyze and forecast the total monthly handle using historical data. I will then estimate the Super Bowl handle by inferring the percentage of February's total handle historically attributable to Super Bowl wagers. This process involves exploring the trends, seasonal patterns, and other key features of the time series data, as well as decomposing and modeling the series to make accurate predictions.

## Data Preparation

```r
# Read in monthly handle data, retrieved from LegalSportsReport
monthly_handle <- read.csv("monthly_handle.csv")
str(monthly_handle)
```

```
## ’data.frame':    76 obs. of  5 variables:
##  $ Month  : chr  "6/1/2018" "7/1/2018" "8/1/2018" "9/1/2018" ...
##  $ Handle : chr  "309,695,659" "294,006,820" "359,275,692" "817,681,310" ...
##  $ Revenue: chr  "24,630,915" "8,433,994" "23,625,881" "95,358,434" ...
##  $ Hold   : num  0.08 0.03 0.07 0.12 0.05 0.06 0.08 0.04 0.06 0.07 ...
##  $ Taxes  : chr  "1,895,011" "926,804" "2,524,419" "12,245,073" ...
```

```r
head(monthly_handle)
```

```
##       Month      Handle    Revenue Hold      Taxes
## 1  6/1/2018 309,695,659 24,630,915 0.08  1,895,011
## 2  7/1/2018 294,006,820  8,433,994 0.03    926,804
## 3  8/1/2018 359,275,692 23,625,881 0.07  2,524,419
## 4  9/1/2018 817,681,310 95,358,434 0.12 12,245,073
## 5 10/1/2018 854,014,077 45,832,391 0.05  5,217,752
## 6 11/1/2018 996,071,884 54,349,323 0.06  6,272,898
```

```
# Convert columns to proper types
monthly_handle <- monthly_handle %>%
  mutate(Month = mdy(Month),
         Mth = substr(Month, 6, 7),
         Handle = as.numeric(gsub(",", "", Handle)),
         Revenue = as.numeric(gsub(",", "", Revenue)),
         Taxes = as.numeric(gsub(",", "", Taxes)))

# Read in monthly data containing number of states with legal sports gambling
monthly_states <- read.csv("monthly-books.csv")
head(monthly_states)
```

```
##            Month Books
## 1      June 2018     3
## 2      July 2018     3
## 3    August 2018     5
## 4 September 2018     5
## 5   October 2018     5
## 6  November 2018     7
```

```
# Convert Months to date type, using dates from monthly_handle
length(monthly_states$Month) == length(monthly_states$Month)
```

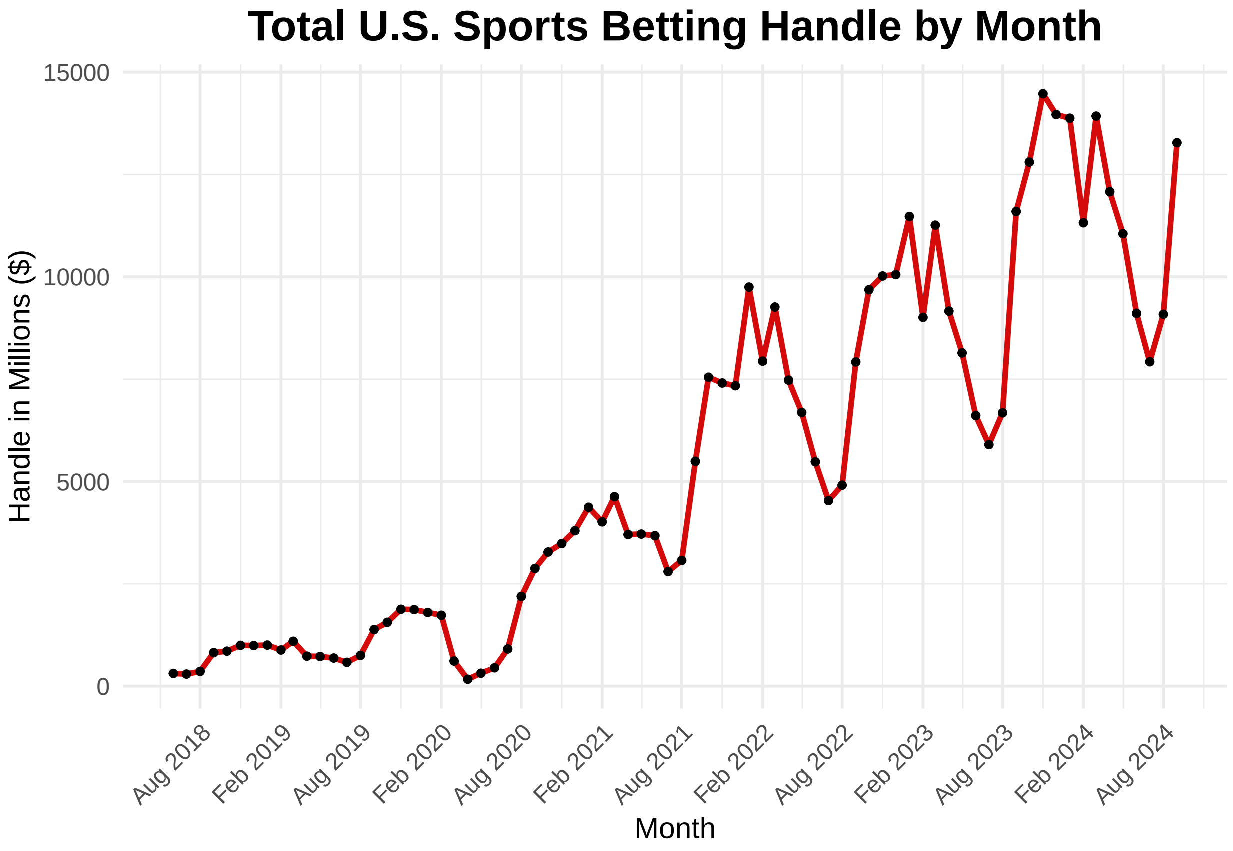
```
## [1] TRUE
```

```
monthly_states <- cbind(monthly_states, monthly_handle$Month)
colnames(monthly_states) <- c("month", "States", "Month")
monthly_states <- monthly_states %>% select(Month, States)

# Join back to monthly_handle
monthly_handle <- monthly_handle %>% left_join(monthly_states, by='Month')

# Create a dataframe storing counts of major American (Big 6) sports in season by month
Mth <- c("01","02","03","04","05","06","07","08","09","10","11","12")
Sports <- c(5,4,3,4,3,3,1,1,3,5,5,5)
monthly_sports <- data.frame(Mth, Sports)

# Join back to monthly handle
monthly_handle <- monthly_handle %>%
  left_join(monthly_sports, by='Mth')

# Add a column for handle in millions for plotting purposes
monthly_handle <- monthly_handle %>% mutate(handle_millions = Handle / 1000000)

head(monthly_handle)
```

```
##        Month    Handle  Revenue Hold    Taxes Mth States Sports handle_millions
## 1 2018-06-01 309695659 24630915 0.08  1895011  06      3      3        309.6957
## 2 2018-07-01 294006820  8433994 0.03   926804  07      3      1        294.0068
## 3 2018-08-01 359275692 23625881 0.07  2524419  08      5      1        359.2757
## 4 2018-09-01 817681310 95358434 0.12 12245073  09      5      3        817.6813
## 5 2018-10-01 854014077 45832391 0.05  5217752  10      5      5        854.0141
## 6 2018-11-01 996071884 54349323 0.06  6272898  11      7      5        996.0719
```

## 1. How Does the Time Series Look?

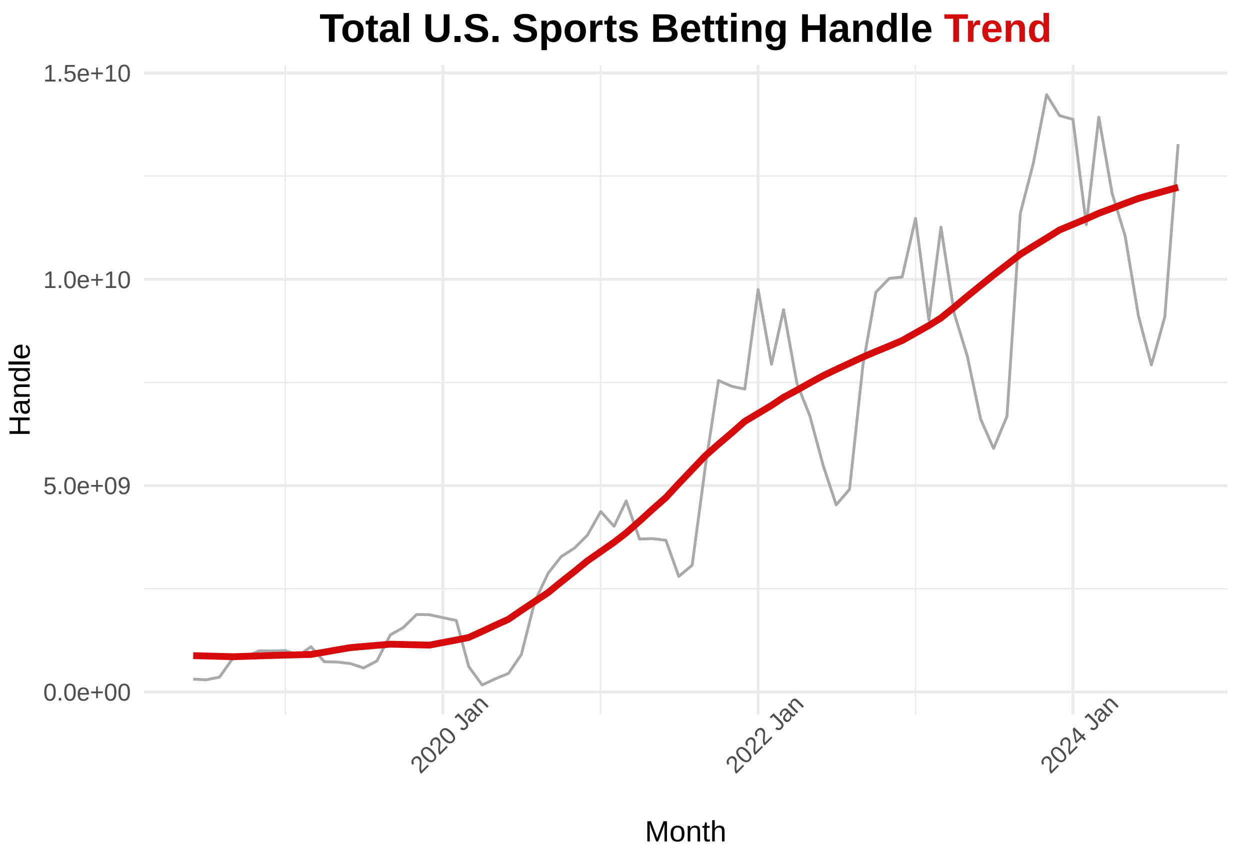
```r
# Plot handle by month
ggplot(monthly_handle, aes(x = Month, y = handle_millions)) +
  geom_line(linewidth = 1, col = '#D50A0A') +
  geom_point(size = 1) +
  labs(
    x = "Month",
    y = "Handle in Millions ($)",
    title = "Total U.S. Sports Betting Handle by Month"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  scale_x_date(
    date_breaks = "6 months",
    date_labels = "%b %Y"
  )
```



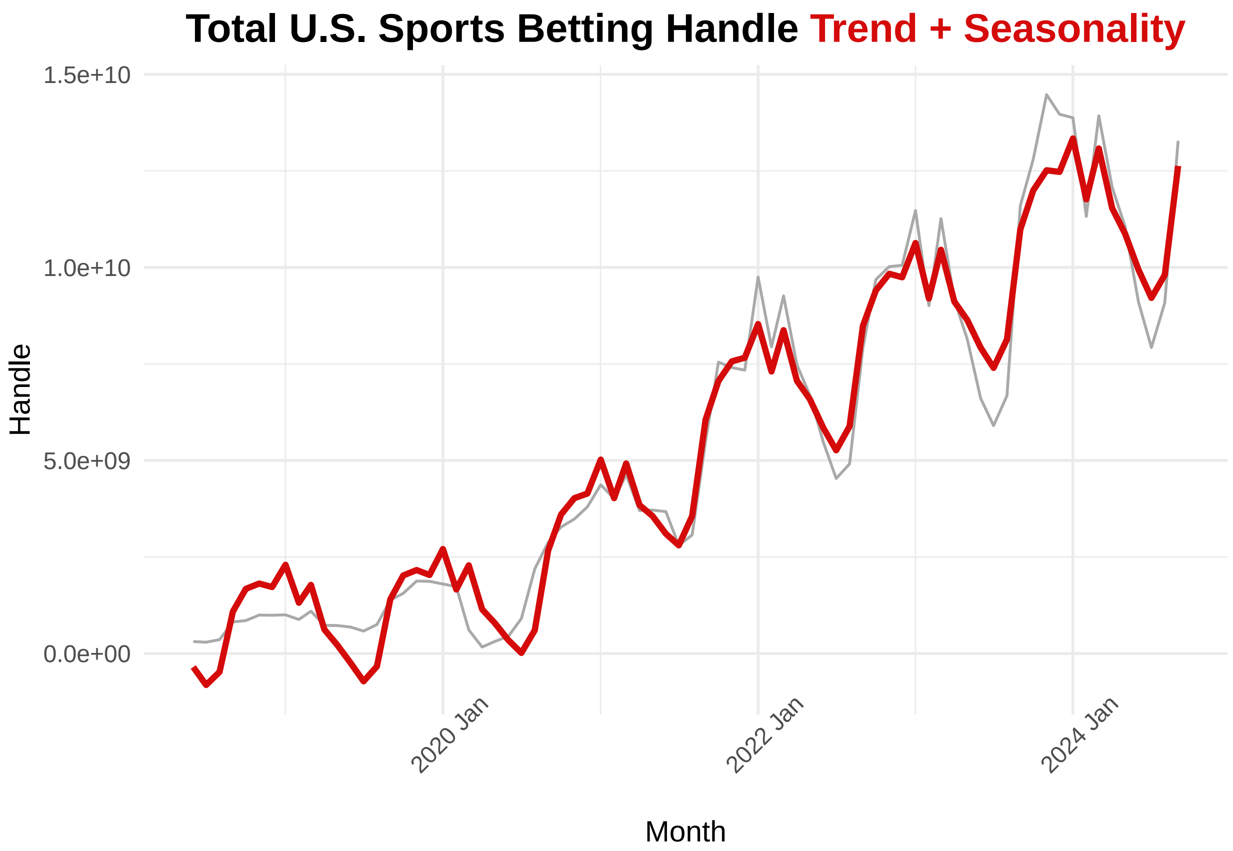
## Observations of the Time Series

- **Upward Trend**: There is a clear overall upward trend in the total U.S. sports betting handle, indicating consistent growth in the market over the years. This may reflect the expansion of legalized sports betting across states and increased public interest.
- **Seasonality**: The sports betting handle exhibits strong seasonal patterns, with peaks around February (likely due to the Super Bowl, which is one of the largest betting events in the U.S.). Significant drops are observed around July, likely due to the relative lack of major sporting events during this time, apart from midseason MLB games.
- **Impact of External Factors**: The dip observed around early 2020 likely reflects the impact of the COVID-19 pandemic, which temporarily halted many sports events. The subsequent recovery in mid to late 2020 coincides with the resumption of sports. This also occurred at a time when the industry was much less mature and there were fewer than 10 states with legal sports betting.
- **Accelerating Growth**: Growth appears to have accelerated around 2021-2022. This could correlate with new states launching legalized sports betting or increased adoption of mobile betting platforms.

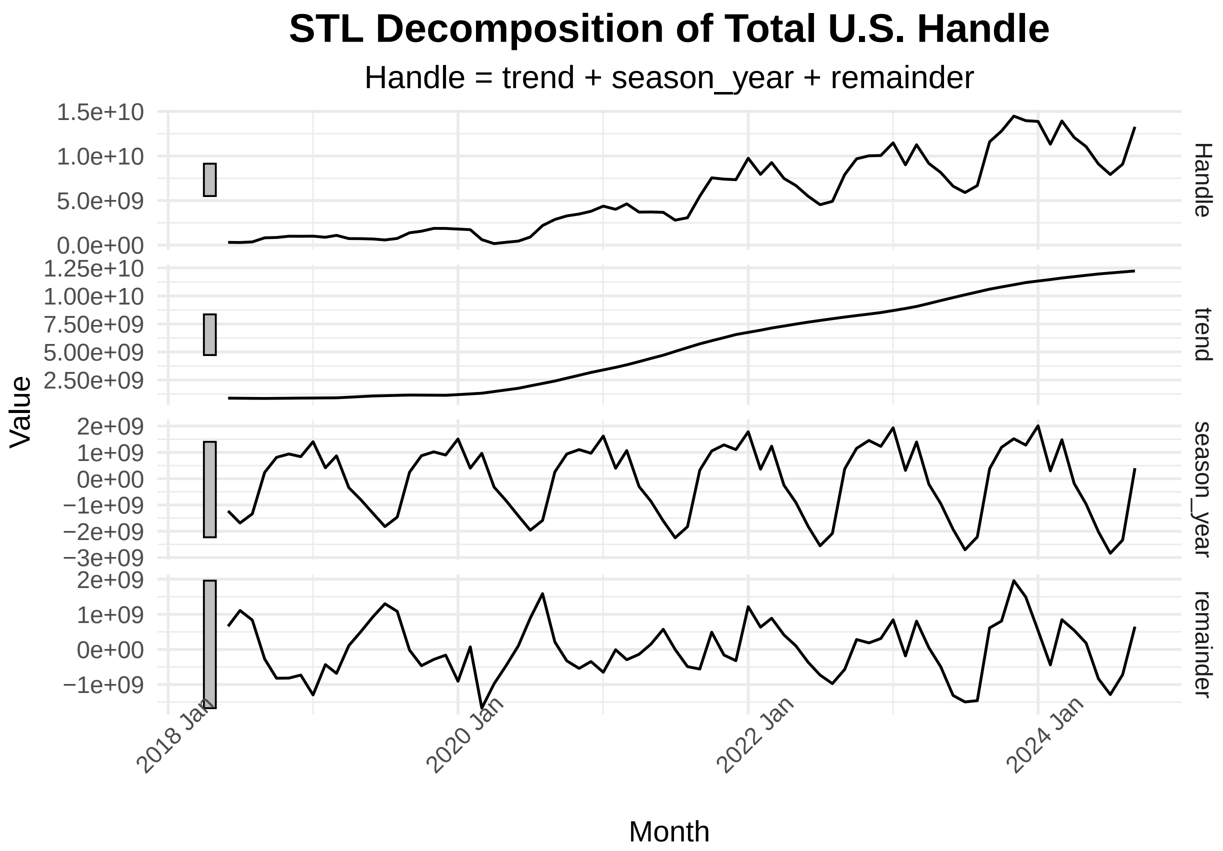
```r
# Create a tsibble
monthly_handle_ts <- monthly_handle %>%
  select(Month, Handle) %>%
  mutate(Month = yearmonth(Month)) %>%
  as_tsibble(index = Month)

# STL Decomposition
handle_stl <- monthly_handle_ts %>%
  model(stl = STL(Handle))

# Plot trend component
trend_plot <- monthly_handle_ts %>%
  autoplot(Handle, color = 'darkgrey') +
  autolayer(components(handle_stl), trend, color='#D50A0A', linewidth=1.2) +
  labs(
    x = 'Month',
    y = "Handle",
    title = "Total U.S. Sports Betting Handle <span style='color:#D50A0A;'>Trend</span>"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  )

# Plot trend & season component
trend_season_plot <- monthly_handle_ts %>%
  autoplot(Handle, color = 'darkgrey') +
  autolayer(components(handle_stl), trend + season_year, color='#D50A0A', linewidth=1.1) +
  labs(
    x = 'Month',
    y = "Handle",
    title = "Total U.S. Sports Betting Handle <span style='color:#D50A0A;'>Trend + Seasonality</span>"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  )

# Plot full STL Decomposition
stl_plot <- handle_stl %>%
  components() %>%
  autoplot() +
  labs(
    x = 'Month',
    y = "Value",
    title = "STL Decomposition of Total U.S. Handle"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    plot.subtitle = element_markdown(hjust = 0.5, size = 12),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  )

# Plot STL by Month
stl_month_plot <- handle_stl %>%
  components() %>%
  gg_subseries(season_year) +
  labs(
    x = 'Month',
    title = "STL Monthly Decomposition of Total U.S. Handle"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    plot.subtitle = element_markdown(hjust = 0.5, size = 12),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  )

print(trend_plot)
```



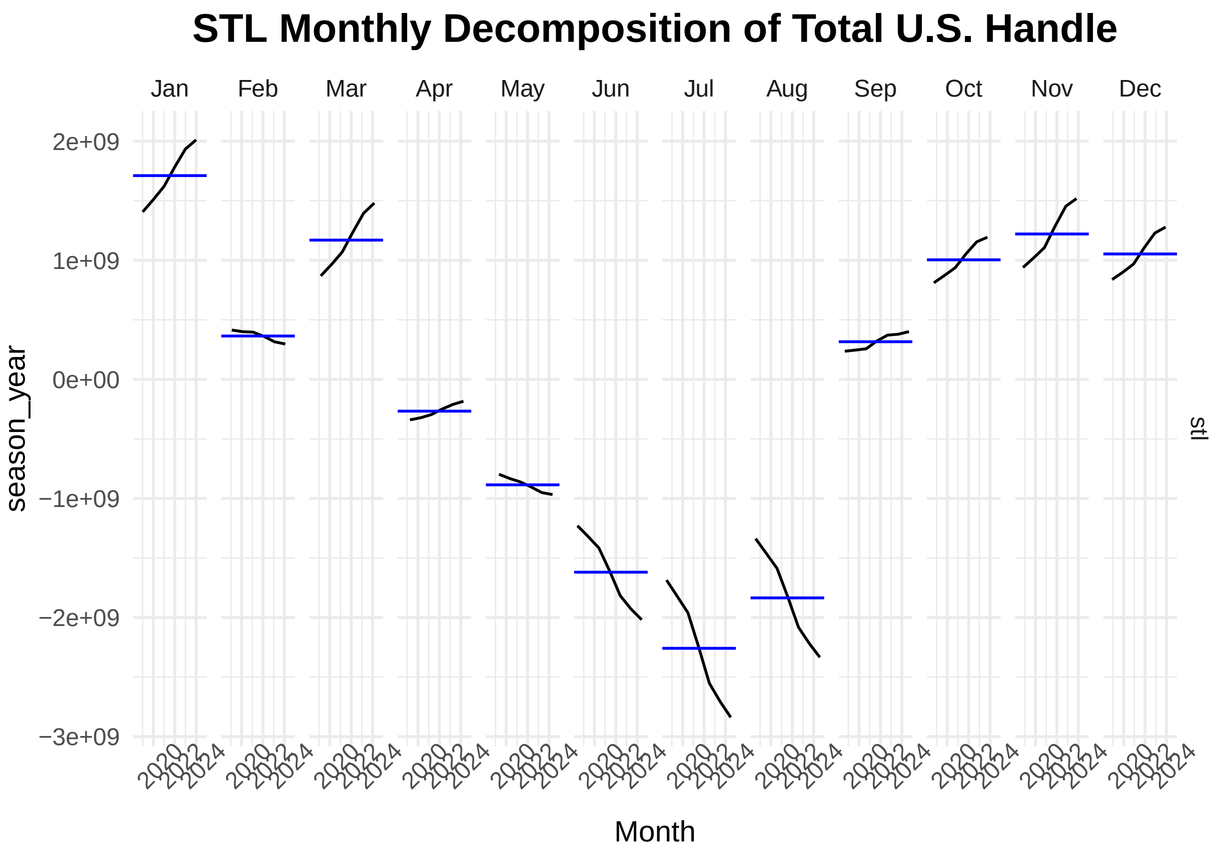
```
print(trend_season_plot)
```



```
print(stl_plot)
```



```r
print(stl_month_plot)
```



## Components of the Time Series

[Plot 1] **Total U.S. Sports Betting Handle Trend**

- **Consistent Upward Growth**: The trend line shows a steady increase in the total U.S. sports betting handle, reflecting the expansion of the sports betting market in the United States.
- **Post-2020 Acceleration**: The trend grows more sharply after 2020, likely coinciding with broader legalization and increased adoption of mobile betting platforms.
- **Market Maturity**: While the trend remains upward, the rate of growth seems to stabilize slightly toward the most recent data points, possibly indicating early signs of market maturity.

[Plot 2] **Total U.S. Sports Betting Handle - Trend + Seasonality**

- **Seasonal Fluctuations**: The addition of seasonality to the trend reveals recurring peaks and troughs, with notable spikes in February and dips in July.
- **Amplified Variability**: Seasonal variability appears to grow alongside the upward trend, with higher peaks and deeper troughs in recent years, indicating increasing dependence on marquee events.
- **Alignment with Sports Calendar**: The seasonal pattern aligns with major sports events, such as March Madness and the start of the NFL season in the fall.

[Plot 3] **STL Decomposition of Total U.S. Handle**

- **Trend Component**: The decomposition confirms the long-term upward trend, with minor deviations likely reflecting policy changes or external shocks (e.g., COVID-19 in 2020).

- **Seasonal Component**: The seasonality is highly regular, with February and March consistently showing positive contributions and July showing the largest negative contributions year over year.
- **Remainder/Noise**: The remainder component captures unexplained variations, which might be tied to irregular events such as state-specific changes in legalization, shifts in betting preferences, or unusual sports outcomes.

[Plot 4] **STL Monthly Decomposition of Total U.S. Handle**

- **Seasonality Breakdown by Month**: This plot further highlights February having the most significant positive seasonal impact, and July having the most negative impact, underscoring the importance of the sports calendar in driving handle.
- **Steady Seasonal Shifts**: The seasonal effects for most months remain consistent across years, indicating stable seasonality unaffected by long-term changes in the trend.
- **Monthly Dependence**: This plot once again highlights the critical importance of specific months (e.g., February, March, and September) in driving the annual handle and emphasizes the dips during the summer months.

## What Other Factors Might Influence the Time Series?

As alluded to above, several factors play a significant role in shaping the total sports betting handle. These include:

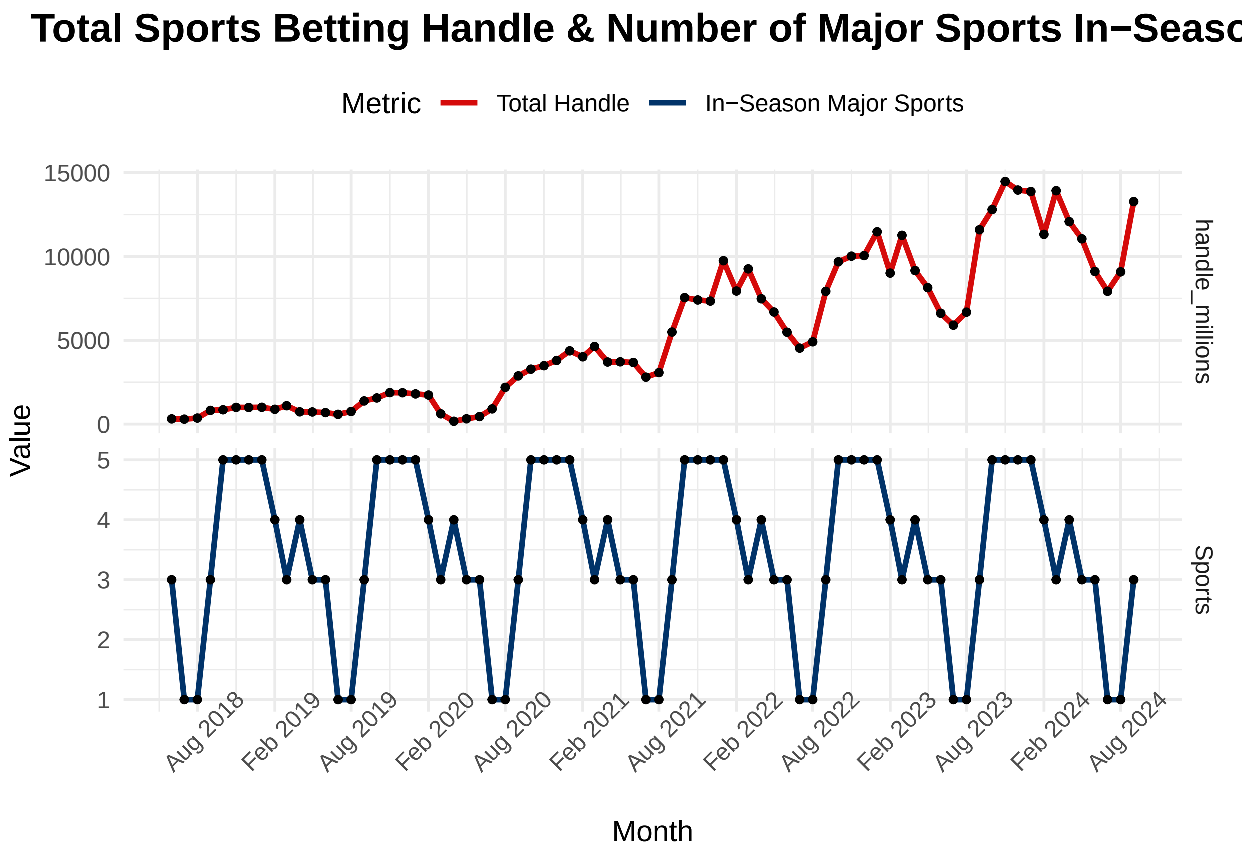
1. **Number of Major Sports in Season**
2. **Marquee Events**
3. **Number of States with Legal Betting**

Marquee events follow the calendars of their respective sports, typically occurring toward the end of a season. Examples include the Super Bowl, March Madness, and the NBA Playoffs, which all drive significant spikes in handle. Both the number of sports in season and the number of states with legal betting will be used as variables in modeling because they both have such strong impacts on the variable of interest, Handle.

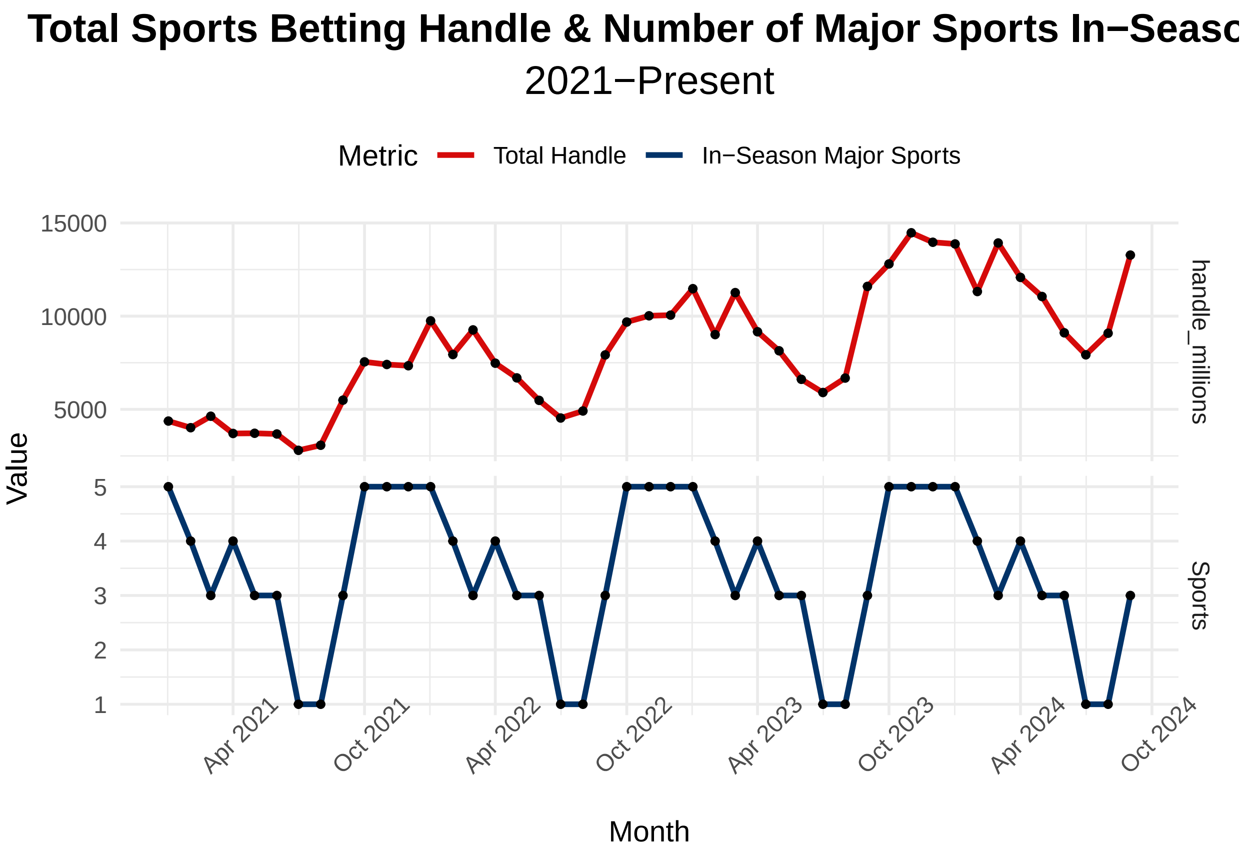
The number of major sports in season aligns closely with the four weather seasons. Handle tends to drop during the summer months (June and July) when only Major League Baseball (MLB) is active. This seasonal lull, often referred to as the “Dog Days of Baseball” in the industry, will likely also be captured by the seasonal component of the model. However, we will also consider adding a dummy variable specifically for July to account for this pronounced drop in activity.

As for the number of states with legal betting, the upward trend in handle appears to be capturing this factor effectively. However, the legalization of sports betting in major states like Texas or California would likely cause a significant jump in handle. While these events could have a profound impact, there is currently no clear timeline or reliable estimate for when these states might legalize sports betting. As a result, this factor will be excluded from the model for now.

```r
# Reshape data to long format
long_data <- monthly_handle %>%
  pivot_longer(cols = c(handle_millions, Sports), names_to = "Metric", values_to = "Value")

# Create faceted plot
handle_sports <- ggplot(long_data, aes(x = Month, y = Value)) +
  geom_line(aes(color = Metric), linewidth = 1) +
  geom_point(size = 1) +
  facet_grid(rows = vars(Metric), scales = "free_y") +
  labs(
    x = "Month",
    y = "Value",
    title = "Total Sports Betting Handle & Number of Major Sports In-Season"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5),
    legend.position = "top"
  ) +
  scale_color_manual(
    values = c("handle_millions" = "#D50A0A", "Sports" = "#013369"),
    labels = c("handle_millions" = "Total Handle", "Sports" = "In-Season Major Sports")
  ) +
  scale_x_date(
    date_breaks = "6 months",
    date_labels = "%b %Y"
  )

# Create faceted plot starting in 2021
handle_sports_2021 <- ggplot(long_data[long_data$Month >= '2021-01-01', ], aes(x = Month, y = Value)) +
  geom_line(aes(color = Metric), linewidth = 1) +
  geom_point(size = 1) +
  facet_grid(rows = vars(Metric), scales = "free_y") +
  labs(
    x = "Month",
    y = "Value",
    title = "Total Sports Betting Handle & Number of Major Sports In-Season",
    subtitle = "2021-Present"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.5, size = 15),
    legend.position = "top"
  ) +
  scale_color_manual(
    values = c("handle_millions" = "#D50A0A", "Sports" = "#013369"),
    labels = c("handle_millions" = "Total Handle", "Sports" = "In-Season Major Sports")
  ) +
  scale_x_date(
    date_breaks = "6 months",
    date_labels = "%b %Y"
  )

print(handle_sports)
```



```r
print(handle_sports_2021)
```



## Number of Major Sports In-Season

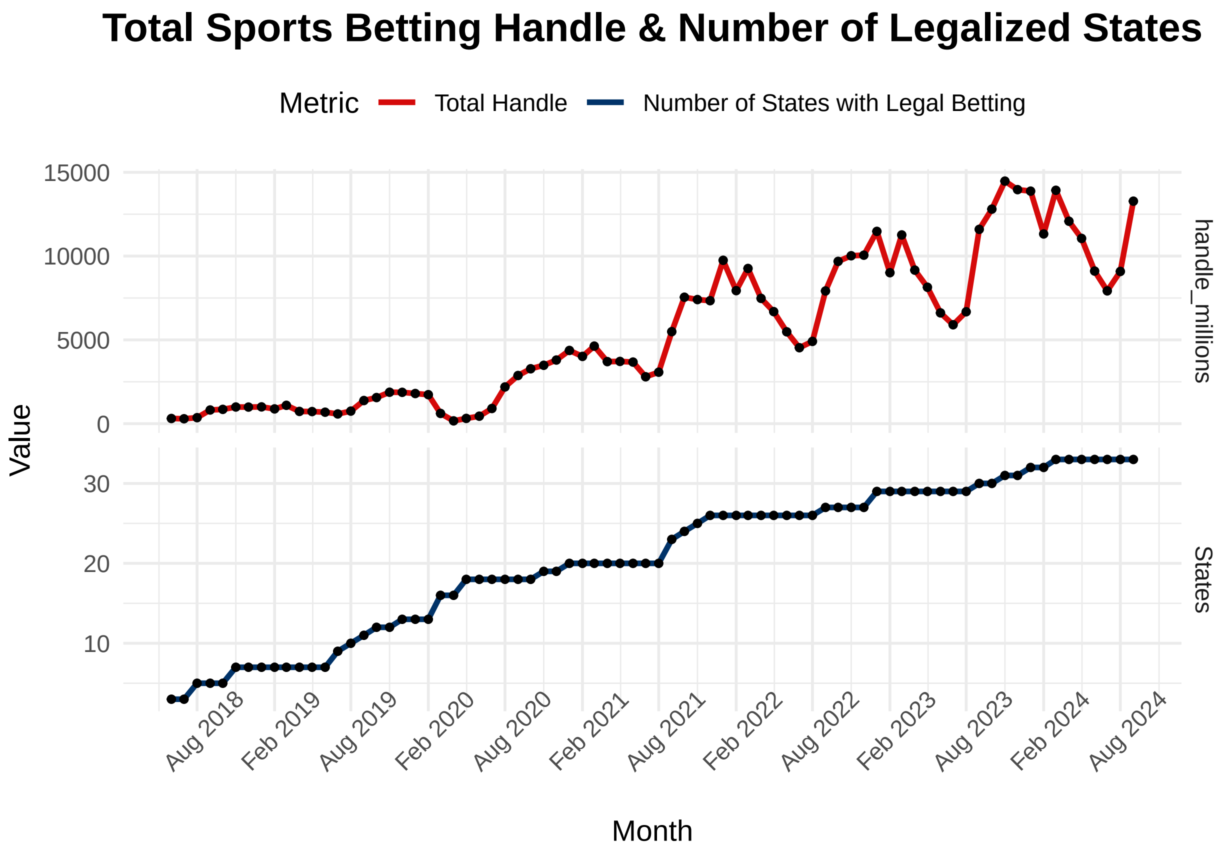
[Plot 1] **Total Sports Betting Handle & Number of Major Sports In-Season**

This plot shows a clear alignment between the number of major sports in season and the total handle, with higher handles occurring during months with more active sports leagues. The seasonal dips in handle, particularly in July, coincide with the lull when only MLB is in season.

[Plot 2] **Total Sports Betting Handle & Number of Major Sports In-Season (2021-Present)**

Focusing on more recent data from a more mature version of the industry, this plot reaffirms the strong relationship between the number of major sports in season and handle. The spikes during fall and winter months highlight the importance of concurrent sports seasons (e.g., NFL, NBA, NHL) in driving betting activity.

```r
# Reshape data to long format
long_df <- monthly_handle %>%
  pivot_longer(cols = c(handle_millions, States), names_to = "Metric", values_to = "Value") %>%
  mutate(Metric = factor(Metric, levels = c("handle_millions", "States")))

# Create faceted plot for handle vs books
states_plot <- ggplot(long_df, aes(x = Month, y = Value)) +
  geom_line(aes(color = Metric), linewidth = 1) +
  geom_point(size = 1) +
  facet_grid(rows = vars(Metric), scales = "free_y") +
  labs(
    x = "Month",
    y = "Value",
    title = "Total Sports Betting Handle & Number of Legalized States",
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5),
    legend.position = "top"
  ) +
  scale_color_manual(
    values = c("handle_millions" = "#D50A0A", "States" = "#013369"),
    labels = c("handle_millions" = "Total Handle", "States" = "Number of States with Legal Betting")
  ) +
  scale_x_date(
    date_breaks = "6 months",
    date_labels = "%b %Y"
  )

print(states_plot)
```



## # Number of States with Legal Sports Betting

This plot highlights the strong correlation between the increase in the number of states with legalized sports betting and the growth in total handle. As more states legalize betting, the handle shows a corresponding upward trend, reflecting the expanding market and accessibility of sports betting across the U.S.

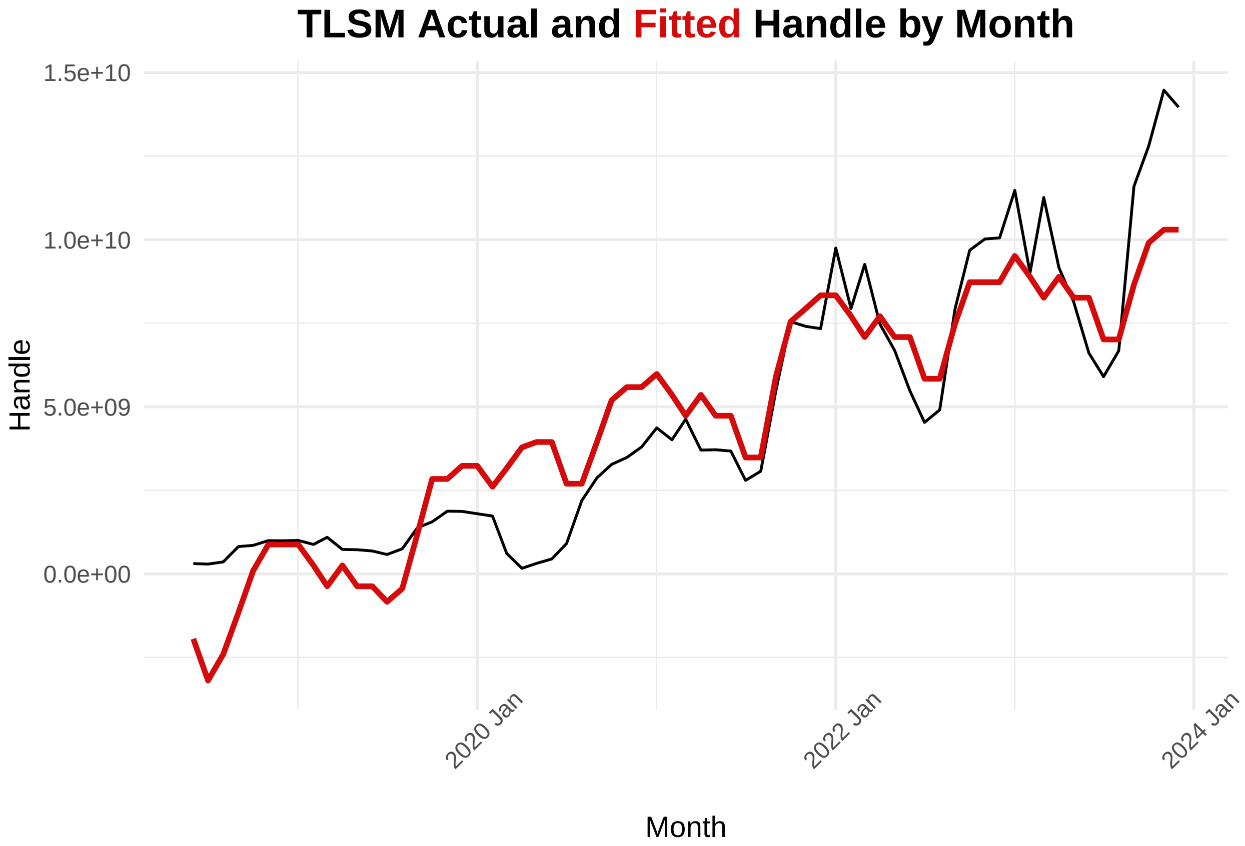
# 3. Modeling the Time Series and Finding the Best Fit

## Model A - Basic Time Series Linear Model

```r
# Remove 2024 data - we will use it later for predictions
tlsm_train <- monthly_handle[monthly_handle$Month < '2024-01-01', ]
tlsm_pred <- monthly_handle[monthly_handle$Month >= '2024-01-01', ]

# Convert to tsibbles
tlsm_train_ts <- tlsm_train %>%
  select(Month, Handle, States, Sports) %>%
  mutate(Month = yearmonth(Month)) %>%
  as_tsibble(index = Month)

tlsm_pred_ts <- tlsm_pred %>%
  select(Month, Handle, States, Sports) %>%
  mutate(Month = yearmonth(Month)) %>%
  as_tsibble(index = Month)

# TLSM Model with only Sports and States
basic_tlsm <- tlsm_train_ts %>%
  model(
    tslm = TSLM(
      Handle ~ Sports + States))

basic_tlsm_plot <- augment(basic_tlsm) %>%
  ggplot(aes(x = Month)) +
  geom_line(aes(y = Handle), color = "black") + # Actual line
  geom_line(aes(y = .fitted), color = "#D50A0A", linewidth = 1) +
  labs(
    y = "Handle",
    title = "TLSM Actual and <span style='color:#D50A0A;'>Fitted</span> Handle by Month"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  )

print(report(basic_tlsm))
```

```
## Series: Handle
## Model: TSLM
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.632e+09 -1.198e+09 -1.238e+08  1.144e+09  4.175e+09
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.992e+09  7.208e+08  -6.926 2.51e-09 ***
## Sports       6.247e+08  1.525e+08   4.096  0.00012 ***
## States       3.925e+08  2.545e+07  15.425  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.764e+09 on 64 degrees of freedom
## Multiple R-squared: 0.8067,  Adjusted R-squared: 0.8007
## F-statistic: 133.6 on 2 and 64 DF, p-value: < 2.22e-16
## # A mable: 1 x 1
##       tslm
##    <model>
## 1  <TSLM>
```

```
print(basic_tlsm_plot)
```



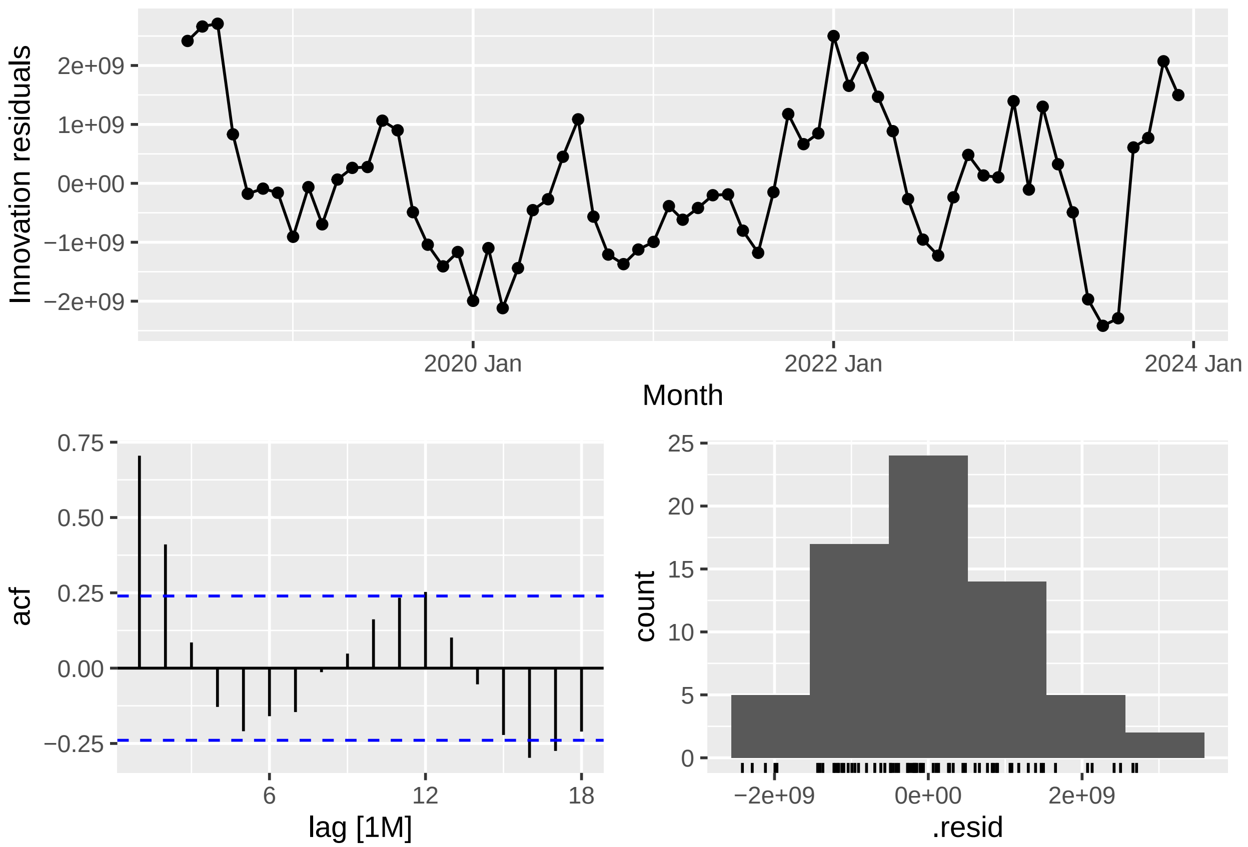
A TLSM model with only 2 features, Sports and States has an R-Square of 0.8, meaning it is capturing 80% of the variance in the time series. Based on the plot, it appears the model is getting the correct direction and idea for both the trend and seasonal components, but missing on the magnitudes.

### Model B - Time Series Linear Model with Trend and Season

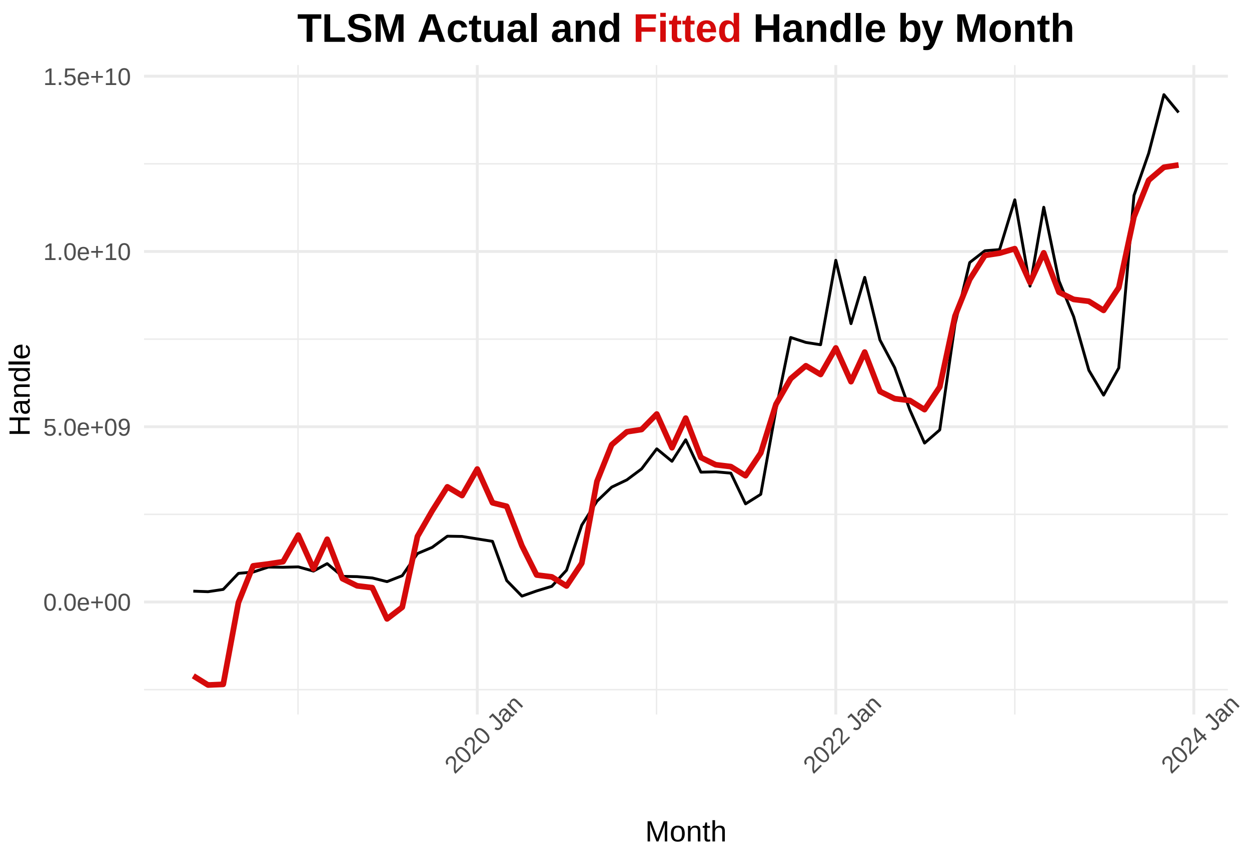
```
# TLSM with trend and season
model_tlsm <- tlsm_train_ts %>%
  model(
    tslm = TSLM(
      Handle ~ Sports + States + trend() + season()))

report(model_tlsm)
```

```
## Series: Handle
## Model: TSLM
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.418e+09 -9.310e+08 -1.598e+08  8.403e+08  2.709e+09
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.639e+09  1.737e+09  -0.944  0.34961
## Sports          6.476e+08  4.177e+08   1.550  0.12699
## States         -3.153e+08  1.098e+08  -2.873  0.00584 **
## trend()         3.147e+08  4.809e+07   6.544 2.44e-08 ***
## season()year2  -6.314e+08  7.401e+08  -0.853  0.39742
## season()year3   5.479e+08  8.324e+08   0.658  0.51326
## season()year4  -1.538e+09  7.408e+08  -2.077  0.04270 *
## season()year5  -1.412e+09  8.301e+08  -1.700  0.09491 .
## season()year6  -1.779e+09  7.938e+08  -2.241  0.02922 *
## season()year7  -1.059e+09  1.394e+09  -0.760  0.45048
## season()year8  -7.257e+08  1.394e+09  -0.521  0.60470
## season()year9          NA         NA      NA       NA
## season()year10 -5.645e+08  8.368e+08  -0.675  0.50287
## season()year11 -1.942e+08  8.328e+08  -0.233  0.81647
## season()year12 -4.439e+08  8.340e+08  -0.532  0.59675
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.37e+09 on 53 degrees of freedom
## Multiple R-squared: 0.9034,  Adjusted R-squared: 0.8797
## F-statistic: 38.13 on 13 and 53 DF, p-value: < 2.22e-16
```

```
residuals_tlsm <- model_tlsm %>% gg_tsresiduals()

plot_tlsm <- augment(model_tlsm) %>%
  ggplot(aes(x = Month)) +
  geom_line(aes(y = Handle), color = "black") + # Actual line
  geom_line(aes(y = .fitted), color = "#D50A0A", linewidth = 1) +
  labs(
    y = "Handle",
    title = "TLSM Actual and <span style='color:#D50A0A;'>Fitted</span> Handle by Month"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  )

print(residuals_tlsm)
```



```r
print(plot_tlsm)
```



The TLSM with trend and season components added is performing much better. This model now has an Adjusted R-Squared of 0.88, a big jump from the prior model. The overall pattern of the fit appears closer to the underlying data, however, looking at the residuals, specifically the autocorrelations, there is room for improvement.

## Model C - Time Series Linear Model with Trend, Season, and Lag

In order to address the high autocorrelations in the TLSM with Trend and Season, we will experiment with adding a Lag, and allowing the model to learn from previous time points.

```
# TLSM with Trend, Season, and Lag
model_tlsm_lag <- tlsm_train_ts %>%
  model(
    tslm = TSLM(Handle ~ trend() + season() + lag(Handle)))

report(model_tlsm_lag)
```

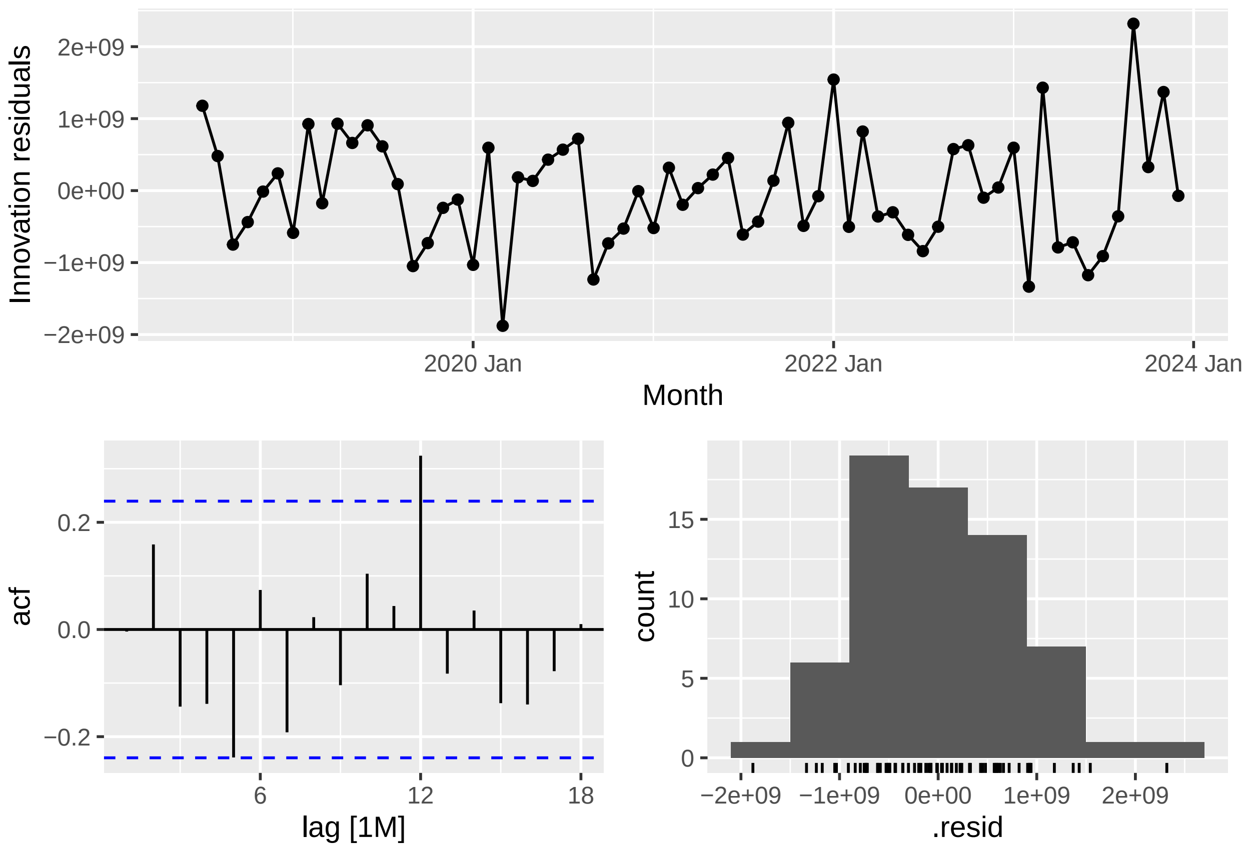
```
## Series: Handle
## Model: TSLM
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.878e+09 -5.259e+08 -4.304e+07  5.749e+08  2.319e+09
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)     4.455e+08  4.336e+08   1.027 0.308985
## trend()         4.648e+07  1.568e+07   2.965 0.004563 **
## season()year2  -1.688e+09  5.526e+08  -3.054 0.003557 **
## season()year3  -3.258e+08  5.509e+08  -0.591 0.556838
## season()year4  -2.008e+09  5.498e+08  -3.653 0.000604 ***
## season()year5  -1.511e+09  5.600e+08  -2.699 0.009367 **
## season()year6  -1.835e+09  5.697e+08  -3.221 0.002202 **
## season()year7  -1.664e+09  5.505e+08  -3.023 0.003876 **
## season()year8  -9.350e+08  5.654e+08  -1.654 0.104231
## season()year9   6.565e+08  5.564e+08   1.180 0.243396
## season()year10 -2.325e+07  5.271e+08  -0.044 0.964986
## season()year11 -3.798e+08  5.279e+08  -0.719 0.475108
## season()year12 -7.952e+08  5.300e+08  -1.500 0.139553
## lag(Handle)     7.789e-01  8.338e-02   9.341 1.04e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 868900000 on 52 degrees of freedom
## Multiple R-squared: 0.9612,  Adjusted R-squared: 0.9515
## F-statistic: 99.03 on 13 and 52 DF, p-value: < 2.22e-16
```

```
tlsm_lag_residuals <- model_tlsm_lag %>% gg_tsresiduals()

tlsm_lag_plot <- augment(model_tlsm_lag) %>%
  ggplot(aes(x = Month)) +
  geom_line(aes(y = Handle), color = "black") +
  geom_line(aes(y = .fitted), color = "#D50A0A", linewidth = 1) +
  labs(
    y = "Handle",
    title = "TLSM With Lag - Actual and <span style='color:#D50A0A;'>Fitted</span> Handle by Month"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  )

print(tlsm_lag_residuals)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```

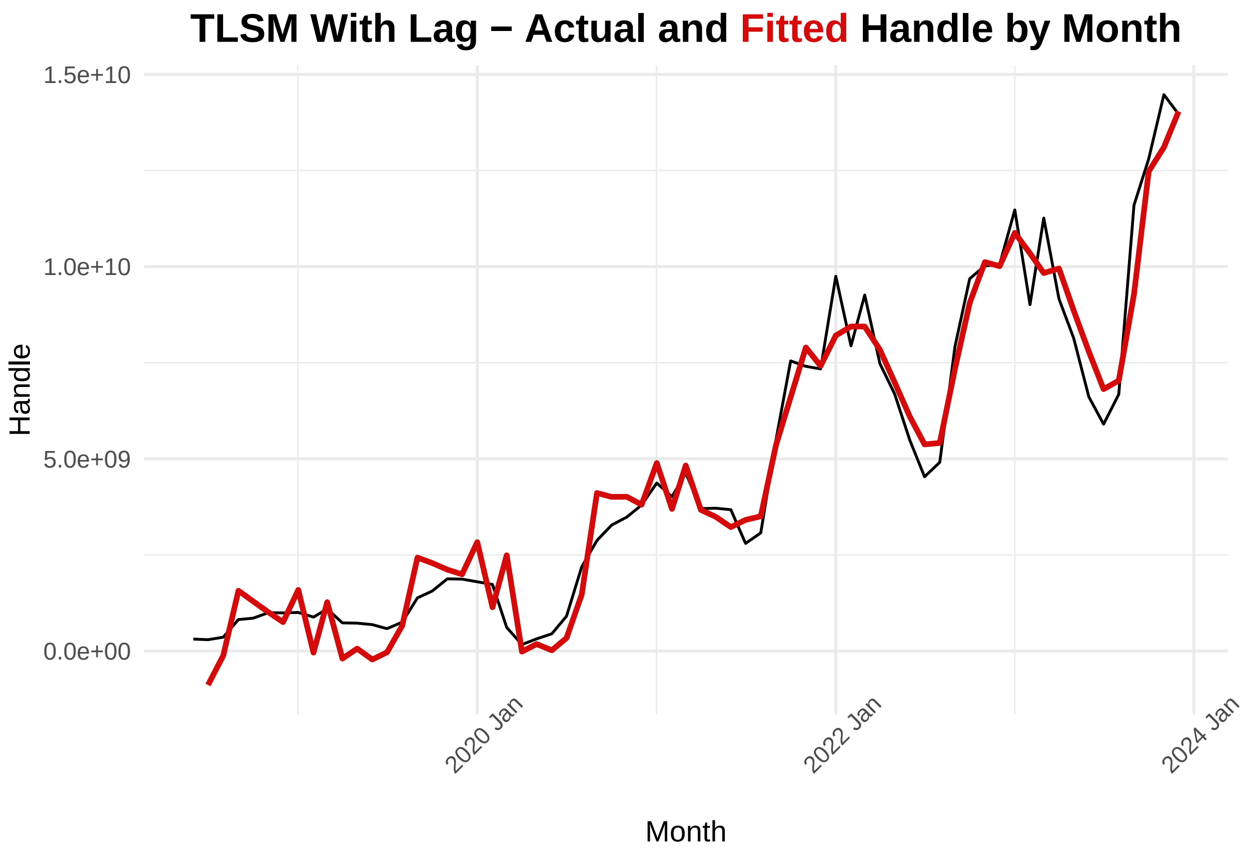
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```



```r
print(tlsm_lag_plot)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (‘geom_line()‘).
```



The TLSM with a season, trend, and lag component is achieving an adjusted R-squared of 0.9515, which is extremely high. The residuals now look more normally distributed. The autocorrelations still do spike around 12 months, but this is a large improvement from the prior model.

From the plotted fit, the model seems to be capturing the trend, season, and peaks very well. It is underestimating some of the troughs but overall looks to fit the data really nicely.

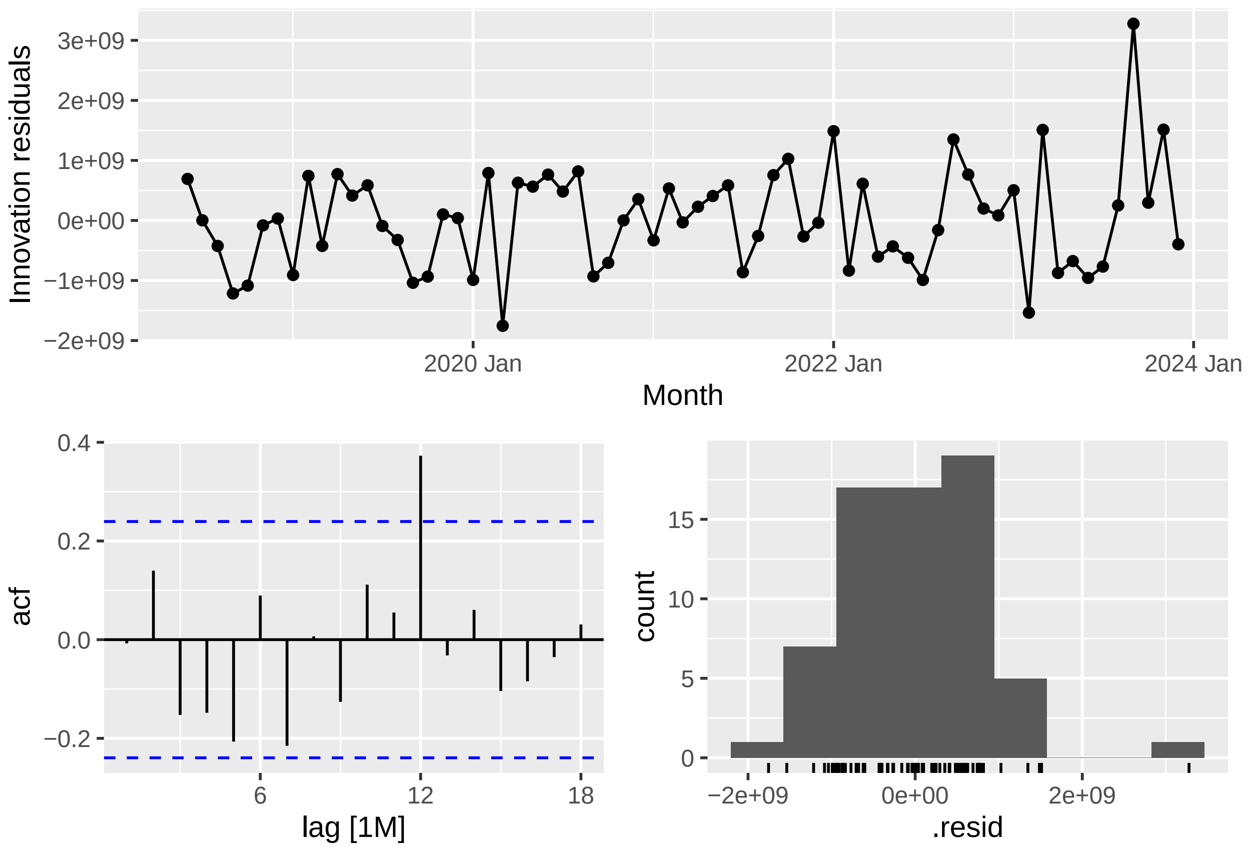
*Note: Both States and Sports were dropped from the model as neither were significant in this version (p > 0.05).*

### Model D - ETS

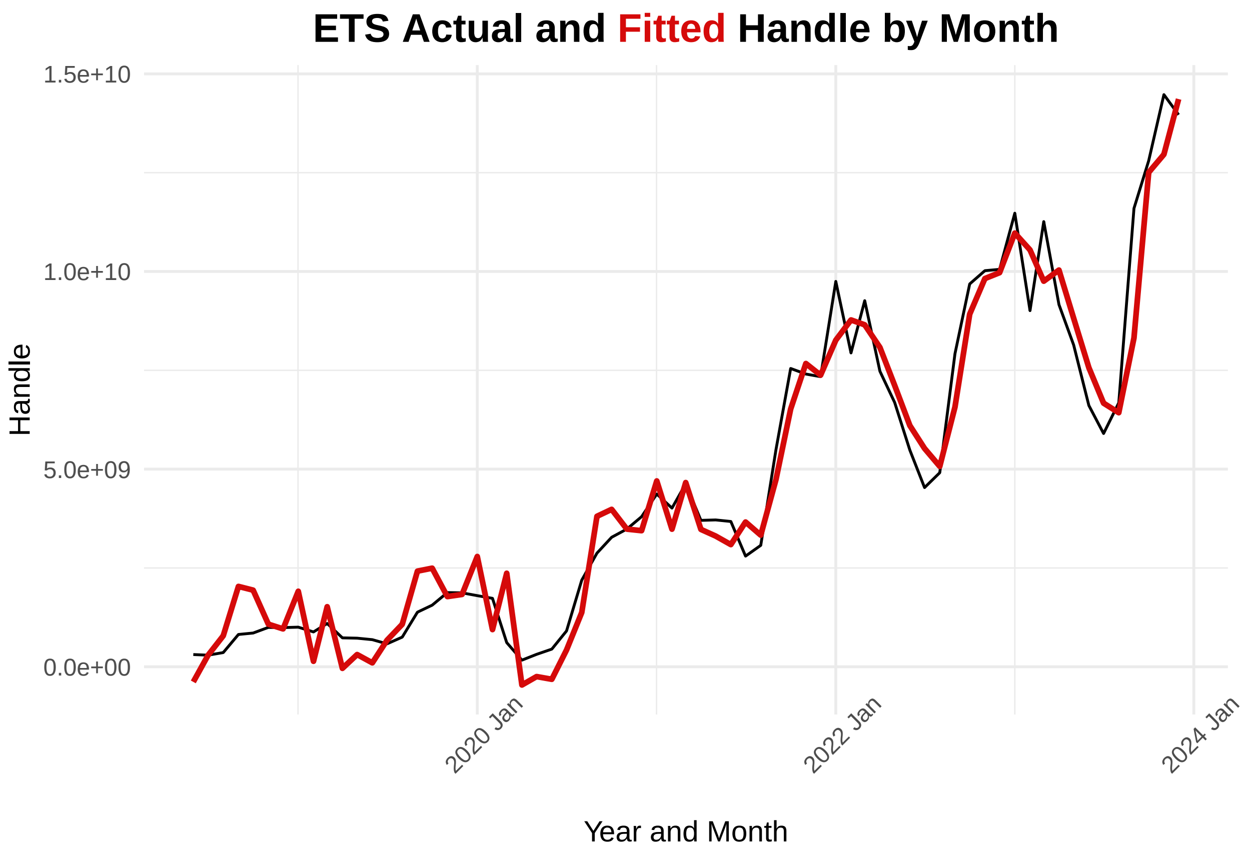
```
# Fit several variations of an ETS model
ets_models <- tlsm_train_ts %>%
  model(
    ANN = ETS(Handle ~ error("A") + trend("N") + season("N")),
    AAN = ETS(Handle ~ error("A") + trend("A") + season("N")),
    AAA = ETS(Handle ~ error("A") + trend("A") + season("A")),
    AAdA = ETS(Handle ~ error("A") + trend("Ad") + season("A"))
)

glance(ets_models) %>% select(.model, AIC, AICc, MSE) %>% arrange(AICc)
```

```
## # A tibble: 4 x 4
##   .model   AIC  AICc     MSE
##   <chr>  <dbl> <dbl>   <dbl>
## 1 AAA    3071. 3084. 7.24e17
## 2 AAdA   3075. 3089. 7.43e17
## 3 ANN    3090. 3091. 1.47e18
## 4 AAN    3093. 3094. 1.43e18
```

```r
# Fit the best ETS model, AAA
ets_fit <- tlsm_train_ts %>%
  model(
    AAA = ETS(Handle ~ error("A") + trend("A") + season("A")))

# Plot best model
ets_plot <- augment(ets_fit) %>%
  ggplot(aes(x = Month)) +
  geom_line(aes(y = Handle, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted"), linewidth = 1) +
  labs(y = "Handle", x = "Year and Month") +
  scale_colour_manual(values = c(Data = "black", Fitted = "#D50A0A")) +
  labs(
    title = "ETS Actual and <span style='color:#D50A0A;'>Fitted</span> Handle by Month"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  ) +
  theme(legend.position="none")

ets_residuals <- ets_fit %>% gg_tsresiduals()

print(ets_residuals)
```



```
print(ets_plot)
```



The best ETS model is the AAA, according the AICc values (AICc = 3083.51). This makes sense because the data does have a strong trend and seasonality, which should be included in a well-fitting model.

This model appears to be a solid fit for the data as well, capturing the overall trend and seasonality, peaks and troughs.

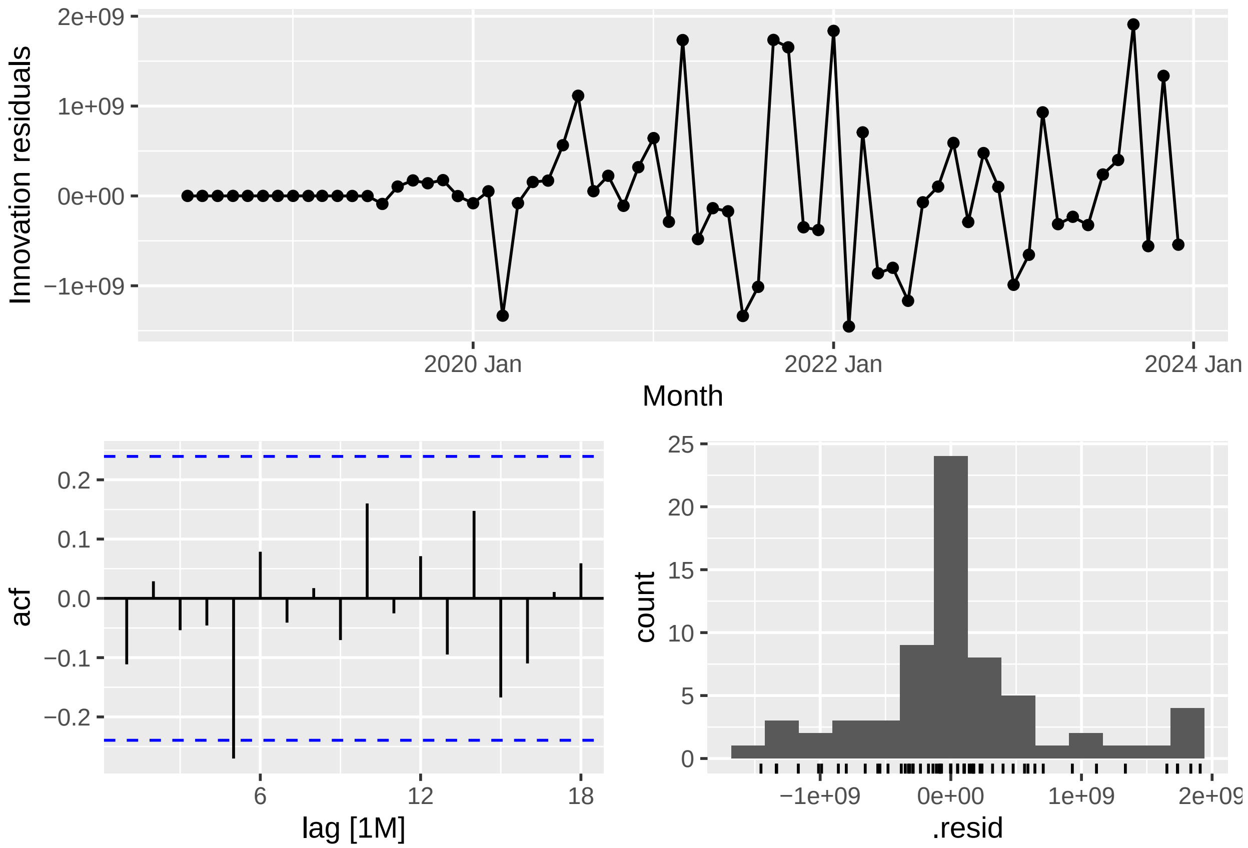
## Model E - ARIMA

```
# Fit a the best SARIMA model
fit_arima <- tlsm_train_ts %>%
  fill_gaps() %>%
  model(arima = ARIMA(Handle))

report(fit_arima)
```

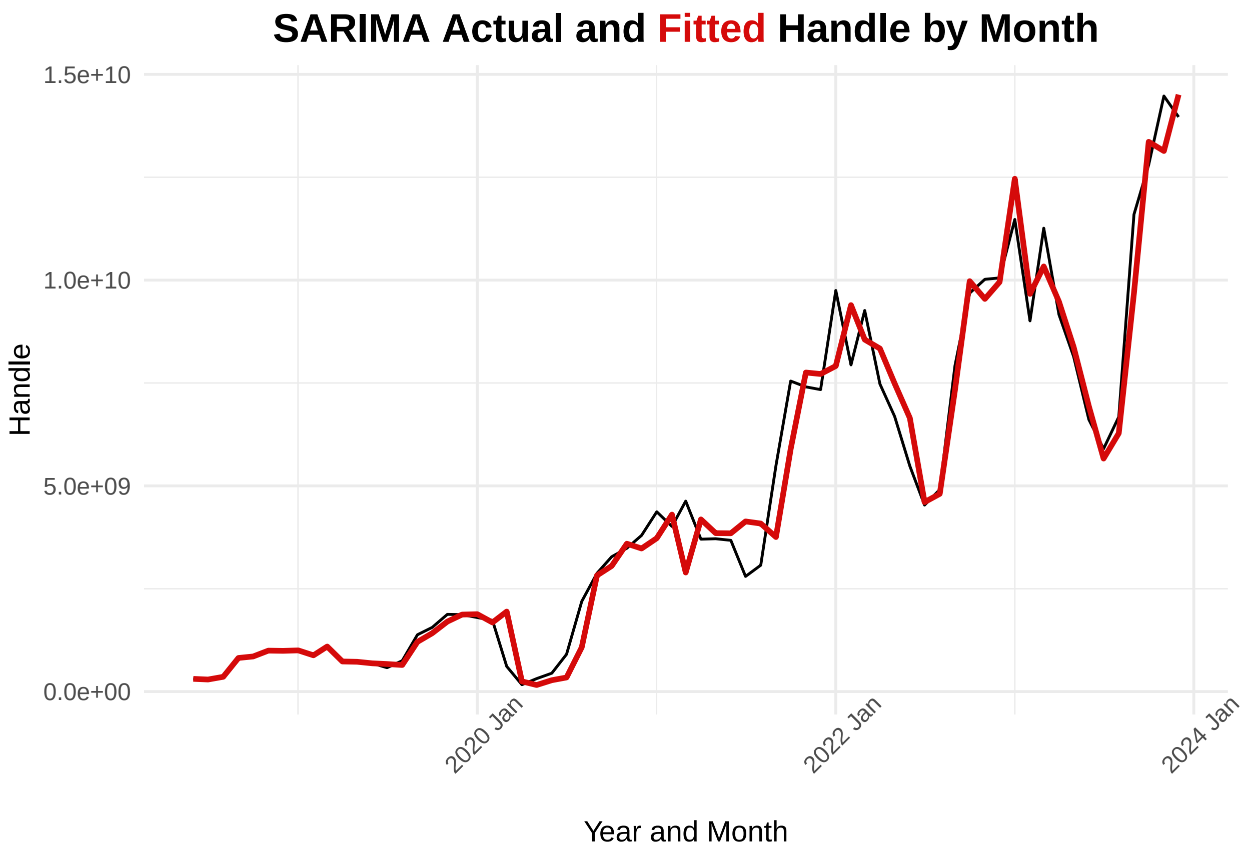
```
## Series: Handle
## Model: ARIMA(0,1,0)(0,1,0)[12]
##
## sigma^2 estimated as 6.386e+17:  log likelihood=-1183.57
## AIC=2369.14   AICc=2369.21   BIC=2371.13
```

```
# Plot best model
arima_plot <- augment(fit_arima) %>%
  ggplot(aes(x = Month)) +
```

```r
  geom_line(aes(y = Handle, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted"), linewidth = 1) +
  labs(y = "Handle", x = "Year and Month") +
  scale_colour_manual(values = c(Data = "black", Fitted = "#D50A0A")) +
  labs(
    title = "SARIMA Actual and <span style='color:#D50A0A;'>Fitted</span> Handle by Month"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  ) +
  theme(legend.position="none")

arima_residuals <- fit_arima %>% gg_tsresiduals()

print(arima_residuals)
```



```r
print(arima_plot)
```



The SARIMA model is showing a really nice fit based on the plot, specially before 2020 and after mid-2022, where it is near-perfect. This model has a corrected AIC value of 2369, which is lower than the one observed from the ETS model.

Of the above models, the **TLSM with Trend, Season, and Lag Components** and the **SARIMA** models both appear to be good fits for data.

- The TLSM captures almost 96% of the variance as indicated by the adjusted R-squared, adequately captures the patterns in the time series as observed in the plot, and has normally-distributed residuals with no noticeable trends.
- The SARIMA model is also performing well, showing a slightly tighter fit than the TLSM based on the plots, similar patterns and observations from the residuals, and a significantly lower AICc than the ETS model. With this model, we can make forecasts beyond the test set, which is not attainable with the TLSM.

For these reasons, we will move forward with the **SARIMA** model.

**SARIMA Model Structure & Components** **Model Structure: ARIMA(0,1,0)(0,1,0)12**

**Non-Seasonal Components (p, d, q):**

- p = 0: No autoregressive (AR) terms.
- d = 1: The data is differenced once to make it stationary (remove trends)
- q = 0: No moving average (MA) terms

**Seasonal Components (P, D, Q):**

- P = 0: No seasonal autoregressive terms.
- D = 1: Seasonal differencing is applied once to remove seasonal patterns
- Q = 0: No seasonal moving average terms.
-

**Metrics**:

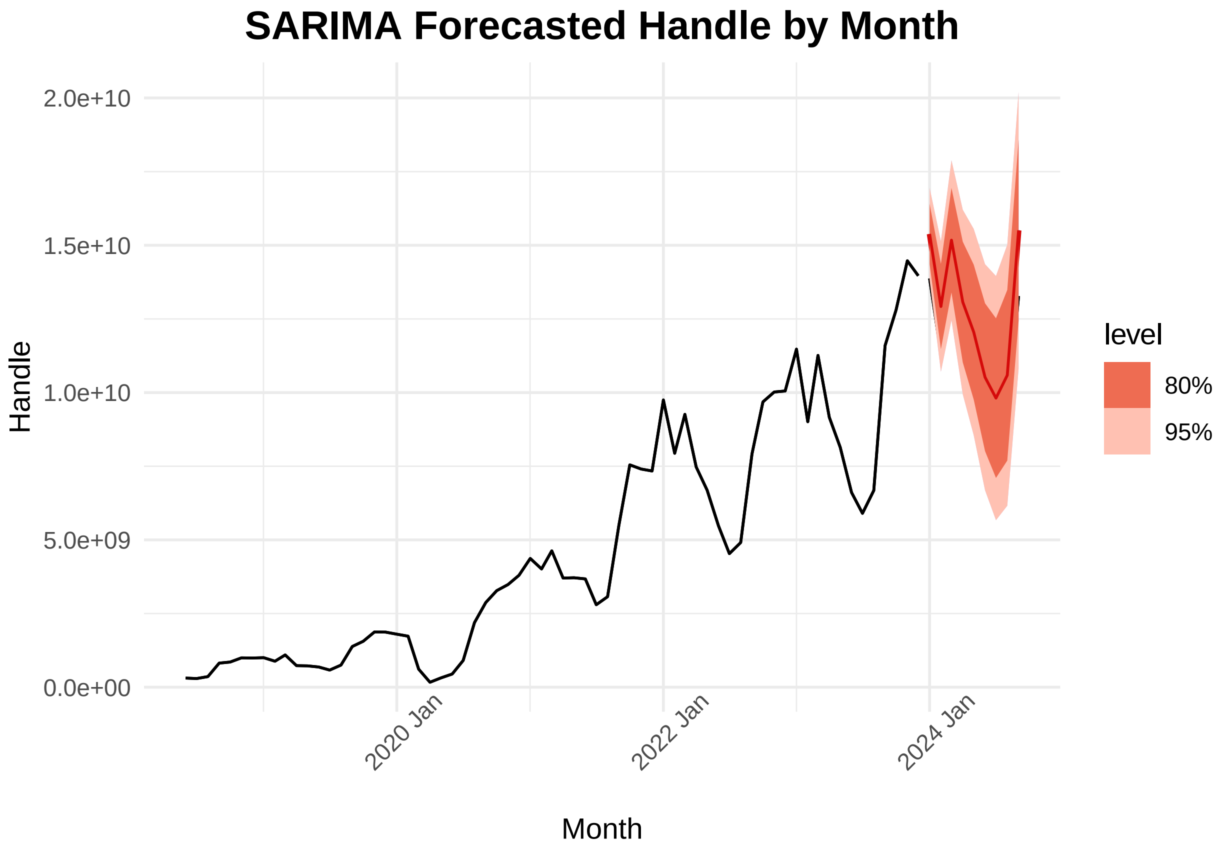
AICc: 2369.21

# 4. Making Forecasts

```
# Forecasting with SARIMA
fit_arima <- tlsm_train_ts %>%
  model(arima = ARIMA(Handle)) %>%
  forecast(h = nrow(tlsm_pred_ts))

arima_forecasts <- fit_arima %>%
  autoplot(tlsm_train_ts) +
  geom_line(data = tlsm_pred_ts, aes(y = Handle), color = "black") +
  geom_line(data = fit_arima, aes(y = .mean), color = "#D50A0A", linewidth=1) +
  theme_minimal() +
  labs(
    y = "Handle",
    title = "SARIMA Forecasted Handle by Month"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  ) +
  autolayer(tlsm_train_ts, series = "Actual", color = "black") +
  autolayer(tlsm_pred_ts, series = "Actual", color = "black") +
  autolayer(fit_arima, series = "Forecast", color = "#D50A0A")
```

```
## Plot variable not specified, automatically selected `.vars = Handle`

## Warning in geom_line(eval_tidy(expr(aes(!!!aes_spec))), data = object, ..., :
## Ignoring unknown parameters: `series`

## Plot variable not specified, automatically selected `.vars = Handle`

## Warning in geom_line(eval_tidy(expr(aes(!!!aes_spec))), data = object, ..., :
## Ignoring unknown parameters: `series`

## Warning in ggdist::geom_lineribbon(without(intvl_mapping, "colour_ramp"), :
## Ignoring unknown parameters: `series`
## Warning in geom_line(mapping = without(mapping, "shape"), data =
## unpack_data(object[single_row[["FALSE"]], : Ignoring unknown parameters:
## ‘series‘

## Scale for fill_ramp is already present.
## Adding another scale for fill_ramp, which will replace the existing scale.
```

```
print(arima_forecasts)
```



# 5. Robustness of the Model & Generalization

The **SARIMA** is the chosen ‘best fit’ model for this data.

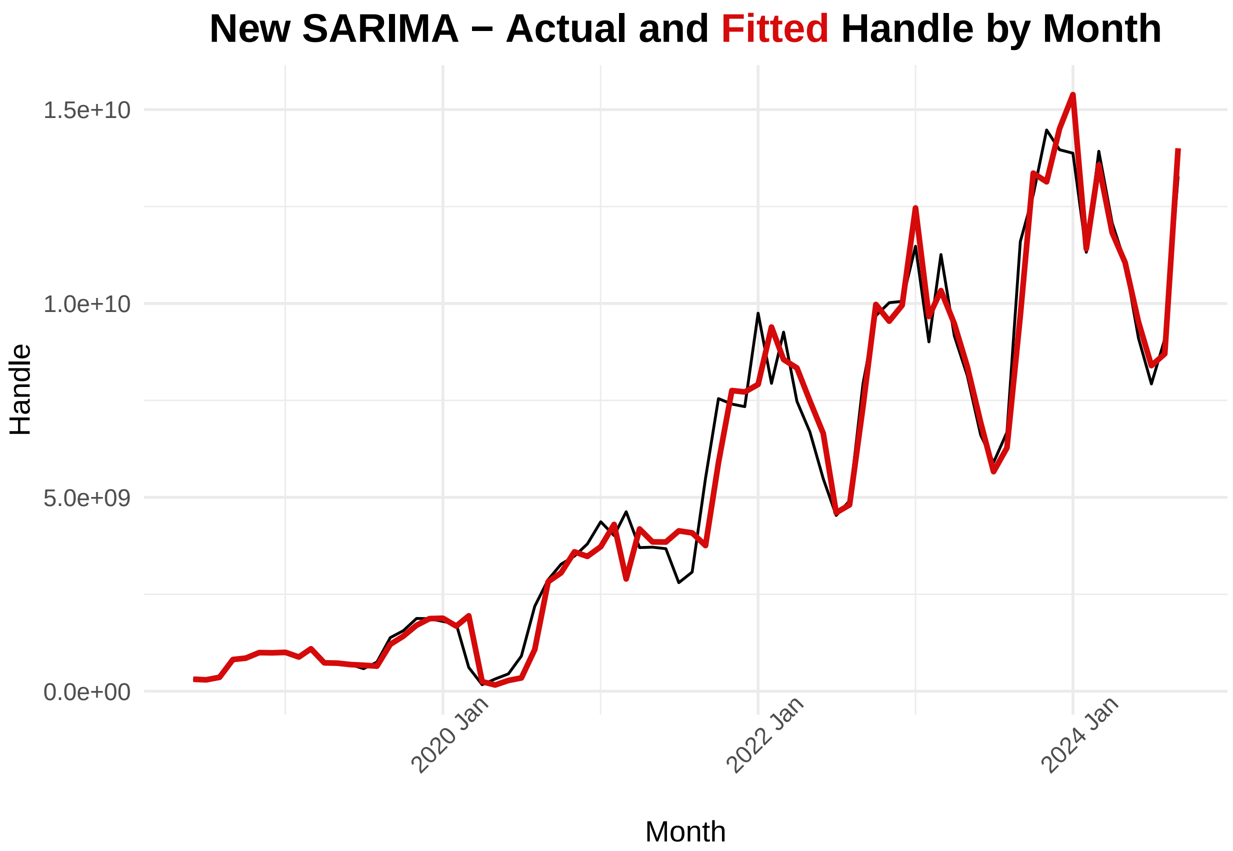
We can test how robust and reliable this model is based reproducibility of the parameters. To do this, we will train the same model on a different subset of the data, and see how well it performs on other periods.

```
# Define new training set, containing all data (including 2024)
tlsm_train2 <- monthly_handle

# Convert to tsibbles
tlsm_train_ts2 <- tlsm_train2 %>%
  select(Month, Handle, States, Sports) %>%
  mutate(Month = yearmonth(Month)) %>%
  as_tsibble(index = Month)
```

```r
# New SARIMA model
fit_sarima <- tlsm_train_ts2 %>%
  fill_gaps() %>%
  model(arima = ARIMA(Handle))

sarima_plot <- augment(fit_sarima) %>%
  ggplot(aes(x = Month)) +
  geom_line(aes(y = Handle), color = "black") +
  geom_line(aes(y = .fitted), color = "#D50A0A", linewidth = 1) +
  labs(
    y = "Handle",
    title = "New SARIMA - Actual and <span style='color:#D50A0A;'>Fitted</span> Handle by Month"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  )

print(report(fit_sarima))
```

```
## Series: Handle
## Model: ARIMA(0,1,0)(0,1,0)[12]
##
## sigma^2 estimated as 6.037e+17:  log likelihood=-1379.06
## AIC=2760.12   AICc=2760.19   BIC=2762.27
## # A mable: 1 x 1
##                       arima
##                     <model>
## 1 <ARIMA(0,1,0)(0,1,0)[12]>
```

```r
print(sarima_plot)
```



### Observations - Robustness

- **Fit**: Based on the plot, this model very closely resembles the earlier one from the SARIMA trained only on data through 2023.
- **Model Structure**: Both models use ARIMA(0,1,0)(0,1,0)12, No AR or MA components are used. Differencing (1,0) and seasonal differencing (1,0)12 are applied.
  - This consistency in model structure shows that the ARIMA configuration is stable across these splits.
- **Evaluation Metrics**: The model trained on all data has higher AIC, AICc, and BIC values, suggesting a worse overall fit compared to the model trained on pre-2024 data.
  - This could be due to the influence of recent data, which may introduce more variability or noise into the model.
- **Variance**: The residual variance is slightly lower for the model trained on all data, suggesting that incorporating the most recent data helps capture some of the variability.

**Reproducibility of Parameters**: The parameters (p, d, q and seasonal counterparts) and the ARIMA structure are identical across splits, indicating high reproducibility in terms of model structure. However, performance metrics (log-likelihood, AIC, BIC) differ, suggesting the fit of the model depends on the training data.

**Robustness of the Model**: The model trained on pre-2024 data performs better based on AIC, AICc, and BIC, which implies a better generalization on older patterns without recent data. The model trained on all data has worse metrics, likely due to variability in recent data not aligning with historical patterns.

### Observations - Generalization

### Strengths of the Model for Generalization

- The ARIMA(0,1,0)(0,1,0)12 model is well-suited to capturing long-term trends (via differencing) and seasonality (with seasonal differencing). This allows the model to generalize well when future data follows historical patterns, especially cyclical or periodic behaviors.
  - The absence of AR and MA components means the model avoids overfitting to short-term fluctuations or noise, making it robust for forecasting where long-term dynamics dominate.
  - The model's structure remained stable across different training splits (random and chronological). This indicates it generalizes well across datasets with similar temporal characteristics.

### Challenges for Generalization

ARIMA models rely on the assumption that future patterns will resemble past ones. Significant deviations, such as regulatory changes (e.g., new states legalizing sports betting) or external shocks (e.g., pandemics), may result in poor forecasts because the model cannot adapt to unforeseen changes.

- The model trained on all data performed slightly worse (higher AIC, AICc, and BIC) than the one trained on pre-2024 data, suggesting that recent data introduces variability. If the recent patterns represent a shift (e.g., rapid market growth), the model may struggle to extrapolate this accurately into the future.
- The model does not incorporate the identified external covariates like the number of states with legal betting, the number of major sports in season, or marquee events (e.g., Super Bowl, March Madness). These factors significantly influence future data but are not explicitly modeled, limiting predictive power.
- The absence of AR or MA terms means the model does not account for short-term autocorrelation, potentially missing important dependencies in consecutive months.

The ARIMA model will generalize well for stable, cyclical trends and regular seasonality but may struggle with sudden changes or external factors not captured in the historical data.

## 6. Projecting Super Bowl Handle

```r
# Forecast out 14 months with original SARIMA to get a projection for February 2025 (Super Bowl)
fit_arima <- tlsm_train_ts %>%
  model(arima = ARIMA(Handle)) %>%
  forecast(h = 14)

arima_forecasts <- fit_arima %>%
  autoplot(tlsm_train_ts) +
  geom_line(data = tlsm_pred_ts, aes(y = Handle), color = "black") +
  geom_line(data = fit_arima, aes(y = .mean), color = "#D50A0A", linewidth=1) +
  theme_minimal() +
  labs(
    y = "Handle",
    title = "SARIMA Forecasted Handle by Month (14 Months)"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_markdown(hjust = 0.5, face = "bold", size = 15),
    axis.text.x = element_text(angle = 45, hjust = 0.5)
  ) +
  autolayer(tlsm_train_ts, series = "Actual", color = "black") +
  autolayer(tlsm_pred_ts, series = "Actual", color = "black") +
  autolayer(fit_arima, series = "Forecast", color = "#D50A0A")
```

```
## Plot variable not specified, automatically selected ‘.vars = Handle‘
```

```
## Warning in geom_line(eval_tidy(expr(aes(!!!aes_spec))), data = object, ..., :
## Ignoring unknown parameters: ‘series‘
```

```
## Plot variable not specified, automatically selected ‘.vars = Handle‘
```

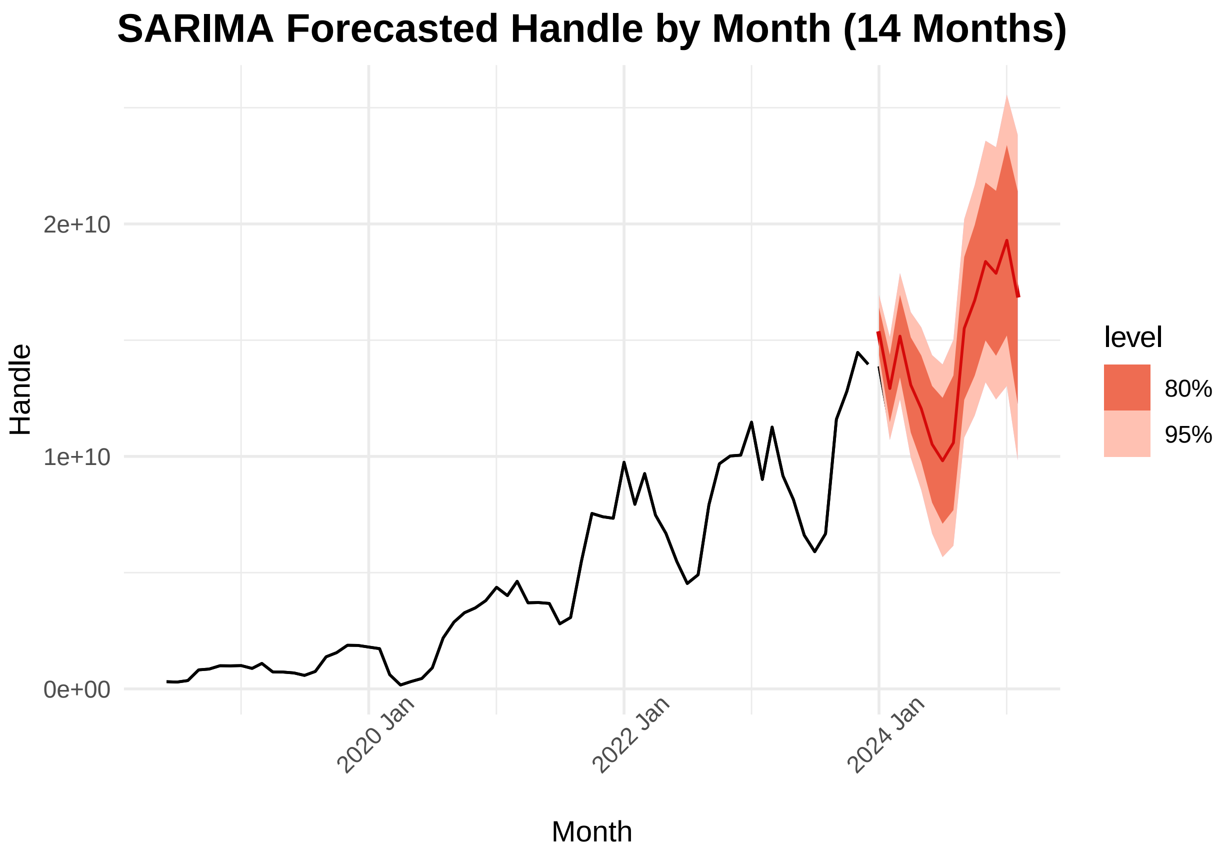
```
## Warning in geom_line(eval_tidy(expr(aes(!!!aes_spec))), data = object, ..., :
## Ignoring unknown parameters: ‘series‘
```

```
## Warning in ggdist::geom_lineribbon(without(intvl_mapping, "colour_ramp"), :
## Ignoring unknown parameters: ‘series‘
```

```
## Warning in geom_line(mapping = without(mapping, "shape"), data =
## unpack_data(object[single_row[["FALSE"]], : Ignoring unknown parameters:
## ‘series‘
```

```
## Scale for fill_ramp is already present.
## Adding another scale for fill_ramp, which will replace the existing scale.
```

```r
print(arima_forecasts)
```



Forecasts appear to be following the trends and patterns in the data, making them very plausible.

```
# Pull projected handle for February
feb <- fit_arima %>% tail(1) %>% pull(.mean)

# 2024 Legal Super Bowl Handle (Sources: Legal Sports Report, Sports Pro Media)
sb_24 <- 1500000000
sb_23 <- 1000000000

# Compute super bowl handle as a percent of total February handle in each year
feb_24 <- monthly_handle %>% filter(Month == '2024-02-01') %>% pull(Handle)
feb_23 <- monthly_handle %>% filter(Month == '2023-02-01') %>% pull(Handle)
pct_24 <- sb_24/feb_24
pct_23 <- sb_23/feb_23

# 2023-24 Super Bowl Handle as a percent of February Handle Growth Rate
rate <- pct_24 - pct_23

# Assuming the same YoY Growth Percentage Growth Rate, project 2025 Super Bowl Handle
percent <- pct_24 + rate
super_bowl_proj <- feb * percent

print(super_bowl_proj)
```

```
## [1] 2592316085
```

We project that Americans will wager a total of **$2.59** Billion dollars (legally) on the Super Bowl in 2025.

This projection underscores the significant growth of the sports betting industry in the United States, highlighting the increasing popularity of legal wagering and the economic impact of marquee events like the Super Bowl. We will check back in in February to determine how close this projection is!