

# A Time Series Approach to Forecasting National Influenza Trends

Team 2: Jun Sik Ryu, Austin Mallie, Cynthia Portales-Loebell

## Table of contents

<b>Introduction</b>	<b>1</b>
<b>Data Import and Cleaning</b>	<b>1</b>
<b>Exploratory Data Analysis</b>	<b>3</b>
<b>Pre-Processing and Data Preparation</b>	<b>10</b>
<b>Train/Test Split</b>	<b>11</b>
<b>Modeling</b>	<b>11</b>

## Introduction

This exploratory data analysis investigates influenza-like illness (ILI) patterns using weekly ILINet data from the CDC.

The goal is to understand seasonal patterns, data quality, and characteristics before modeling.

## Data Import and Cleaning

```
# Import national ILINet data
ili_raw <- ilinet(region = "national")
```

```
glimpse(ili_raw)
```

```
Rows: 1,470
Columns: 16
$ region_type      <chr> "National", "National", "National", "National", "Nati~
$ region          <chr> "National", "National", "National", "National", "Nati~
$ year            <int> 1997, 1997, 1997, 1997, 1997, 1997, 1997, 1997, ~
$ week             <int> 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 5~
$ weighted_ili    <dbl> 1.10148, 1.20007, 1.37876, 1.19920, 1.65618, 1.41326, ~
$ unweighted_ili  <dbl> 1.216860, 1.280640, 1.239060, 1.144730, 1.261120, 1.2~
$ age_0_4          <dbl> 179, 199, 228, 188, 217, 178, 294, 288, 268, 299, 346~
$ age_25_49        <dbl> NA, N~
$ age_25_64        <dbl> 157, 151, 153, 193, 162, 148, 240, 293, 206, 282, 268~
$ age_5_24          <dbl> 205, 242, 266, 236, 280, 281, 328, 456, 343, 415, 388~
$ age_50_64        <dbl> NA, N~
$ age_65            <dbl> 29, 23, 34, 36, 41, 48, 70, 63, 69, 102, 81, 59, 113, ~
$ ilitotal         <dbl> 570, 615, 681, 653, 700, 655, 932, 1100, 886, 1098, 1~
$ num_of_providers <dbl> 192, 191, 219, 213, 213, 195, 248, 256, 252, 253, 242~
$ total_patients   <dbl> 46842, 48023, 54961, 57044, 55506, 51062, 64463, 6674~
$ week_start       <date> 1997-09-28, 1997-10-05, 1997-10-12, 1997-10-19, 1997~
```

```
head(ili_raw)
```

```
# A tibble: 6 x 16
  region_type region  year  week weighted_ili unweighted_ili age_0_4 age_25_49
  <chr>       <chr>   <int> <int>      <dbl>        <dbl>     <dbl>      <dbl>
1 National    National  1997    40      1.10        1.22      179       NA
2 National    National  1997    41      1.20        1.28      199       NA
3 National    National  1997    42      1.38        1.24      228       NA
4 National    National  1997    43      1.20        1.14      188       NA
5 National    National  1997    44      1.66        1.26      217       NA
6 National    National  1997    45      1.41        1.28      178       NA
# i 8 more variables: age_25_64 <dbl>, age_5_24 <dbl>, age_50_64 <dbl>,
#   age_65 <dbl>, ilitotal <dbl>, num_of_providers <dbl>, total_patients <dbl>,
#   week_start <date>
```

Convert to tsibble

```
names(ili_raw)
```

```

[1] "region_type"      "region"          "year"           "week"
[5] "weighted_ili"    "unweighted_ili"   "age_0_4"        "age_25_49"
[9] "age_25_64"        "age_5_24"         "age_50_64"     "age_65"
[13] "ilittotal"       "num_of_providers" "total_patients" "week_start"

ili_ts <- ili_raw |>
  mutate(week = yearweek(week_start)) |>
  as_tsibble(index = week)

ili_ts

# A tsibble: 1,470 x 16 [1W]
# ... with variables:
#   region_type <chr>, region <chr>, year <int>, week <week>,
#   weighted_ili <dbl>, unweighted_ili <dbl>, age_0_4 <dbl>,
#   ...
# i 1,460 more rows
# i 9 more variables: age_25_49 <dbl>, age_25_64 <dbl>, age_5_24 <dbl>,
#   age_50_64 <dbl>, age_65 <dbl>, ilittotal <dbl>, num_of_providers <dbl>,
#   total_patients <dbl>, week_start <date>

```

## Exploratory Data Analysis

```
ili_ts |> has_gaps()
```

```
# A tibble: 1 x 1
  .gaps
  <lgl>
1 FALSE
```

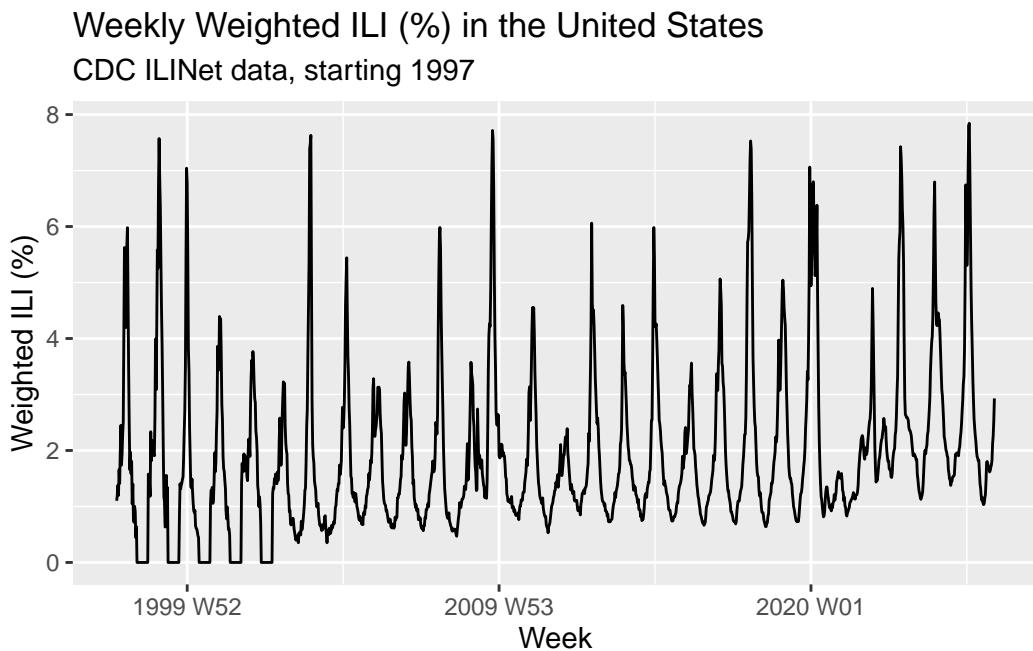
```

#Check for missing ILI values
ili_ts |> filter(is.na(weighted_ili))

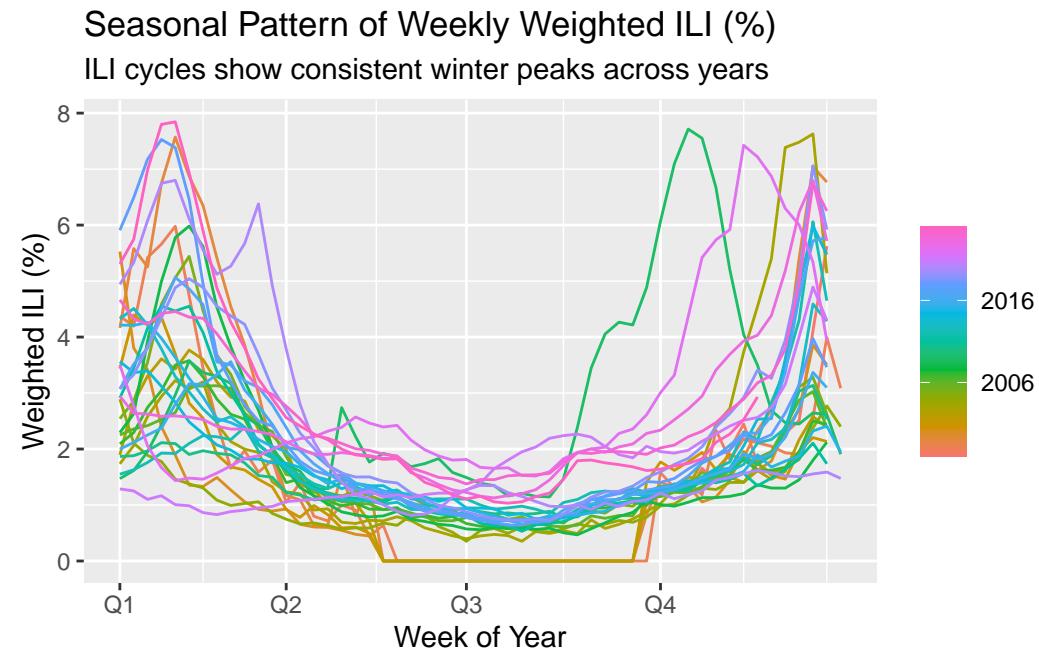
# A tsibble: 0 x 16 [?]
# i 16 variables: region_type <chr>, region <chr>, year <int>, week <week>,
#   weighted_ili <dbl>, unweighted_ili <dbl>, age_0_4 <dbl>, age_25_49 <dbl>,
#   age_25_64 <dbl>, age_5_24 <dbl>, age_50_64 <dbl>, age_65 <dbl>,
#   ilitotal <dbl>, num_of_providers <dbl>, total_patients <dbl>,
#   week_start <date>

# Plot the series
ili_ts |>
  ggplot(aes(x = week, y = weighted_ili)) +
  geom_line() +
  labs(
    title = "Weekly Weighted ILI (%) in the United States",
    subtitle = "CDC ILINet data, starting 1997",
    x = "Week",
    y = "Weighted ILI (%)"
  )

```



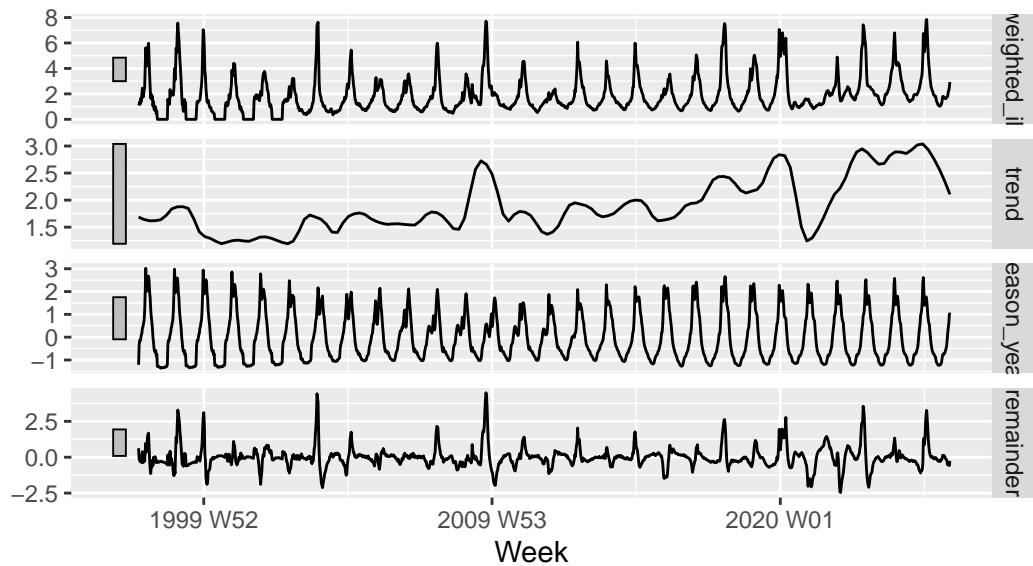
```
# Seasonal plot
ili_ts |>
  gg_season(weighted_ili) +
  labs(
    title = "Seasonal Pattern of Weekly Weighted ILI (%)",
    subtitle = "ILI cycles show consistent winter peaks across years",
    x = "Week of Year",
    y = "Weighted ILI (%)"
  )
```



```
# STL decomposition
ili_ts |>
  model(STL(weighted_ili)) |>
  components() |>
  autoplot() +
  labs(
    title = "STL Decomposition of Weekly Weighted ILI (%)",
    subtitle = "Shows seasonal, trend, and remainder components",
    x = "Week",
    y = NULL
  )
```

## STL Decomposition of Weekly Weighted ILI (%)

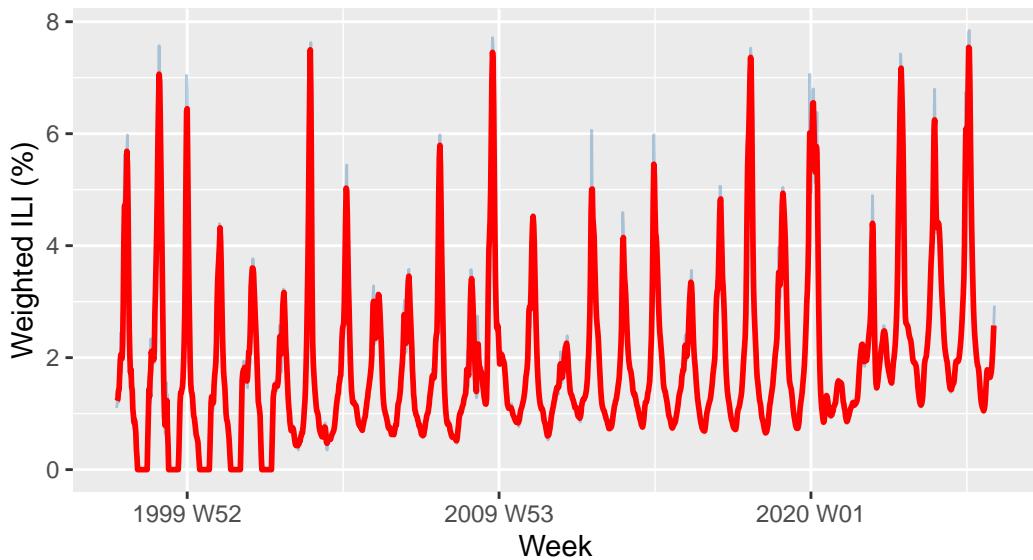
Shows seasonal, trend, and remainder components



```
# Plot a 3 week centered moving average
ili_ts |>
  mutate(
    ma_3 = slider::slide_dbl(
      weighted_ili,
      mean,
      .before = 1,
      .after = 1,
      .complete = TRUE
    )
  ) |>
  ggplot(aes(x = week)) +
  geom_line(aes(y = weighted_ili), alpha = 0.4, color = "steelblue") +
  geom_line(aes(y = ma_3), color = "red", size = 1) +
  labs(
    title = "3-Week Centered Moving Average of Weighted ILI (%)",
    subtitle = "Smoothing highlights the underlying influenza seasonal pattern",
    x = "Week",
    y = "Weighted ILI (%)"
  )
```

### 3-Week Centered Moving Average of Weighted ILI (%)

Smoothing highlights the underlying influenza seasonal pattern



```
# Summary statistics
ili_ts |>
  as_tibble() |>
  summarise(
    min = min(weighted_ili),
    q1 = quantile(weighted_ili, 0.25),
    median = median(weighted_ili),
    mean = mean(weighted_ili),
    q3 = quantile(weighted_ili, 0.75),
    max = max(weighted_ili)
  )
```

```
# A tibble: 1 x 6
  min     q1   median   mean     q3     max
  <dbl> <dbl>   <dbl> <dbl> <dbl> <dbl>
1     0  0.980   1.49  1.90  2.39  7.84
```

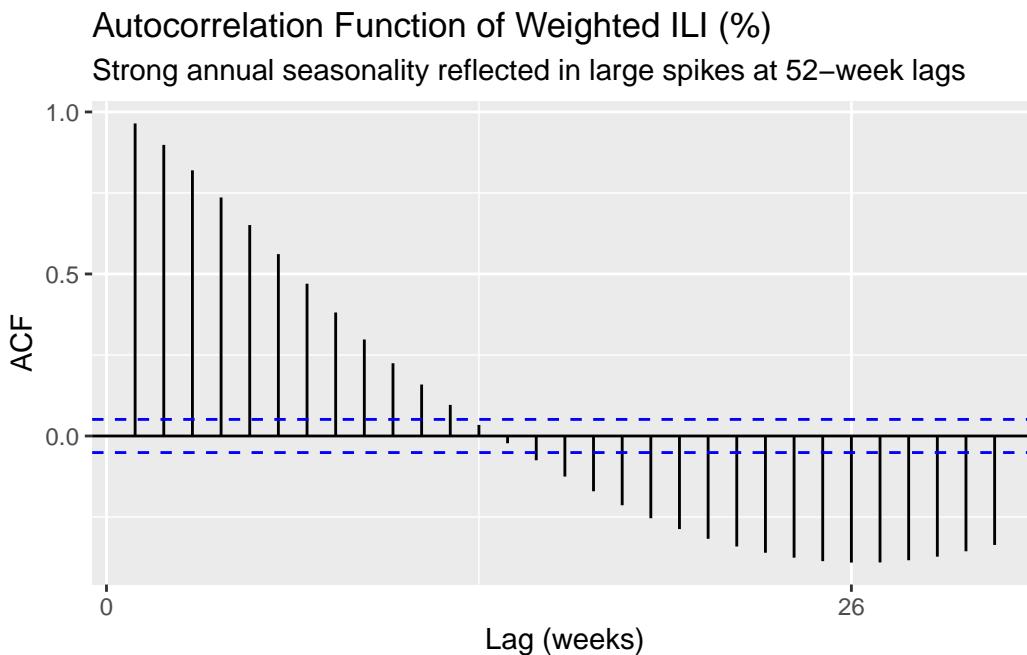
```
# Autocorrelation Function (ACF) of Weekly Weighted ILI (%)

ili_ts |>
  ACF(weighted_ili) |>
  autoplot() +
  labs(
```

```

title = "Autocorrelation Function of Weighted ILI (%)",
subtitle = "Strong annual seasonality reflected in large spikes at 52-week lags",
x = "Lag (weeks)",
y = "ACF"
)

```



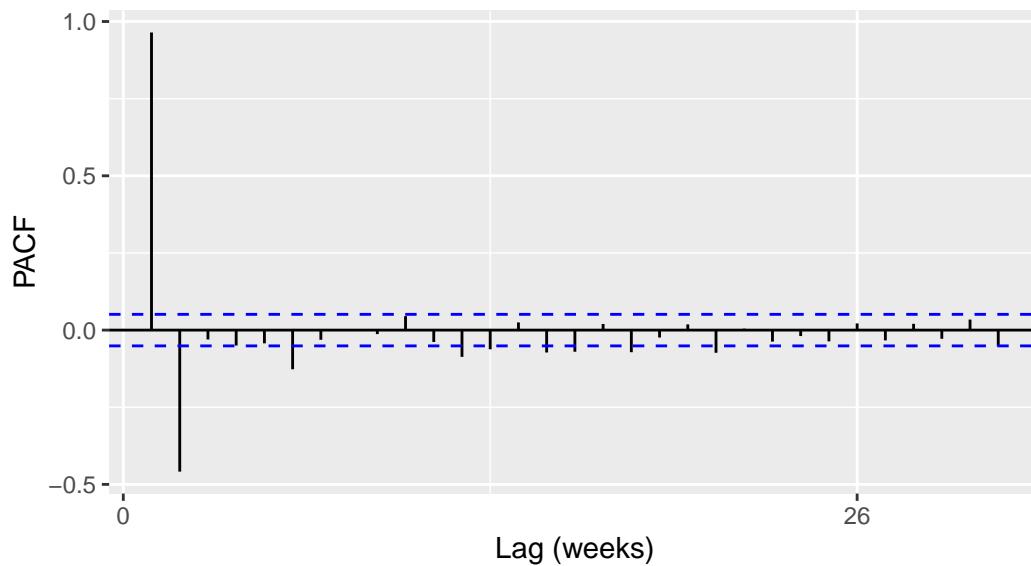
```

# Partial Autocorrelation Function (PACF)
ili_ts |>
  PACF(weighted_ili) |>
  autoplot() +
  labs(
    title = "Partial Autocorrelation Function of Weighted ILI (%)",
    subtitle = "PACF highlights autoregressive behavior and seasonal effects",
    x = "Lag (weeks)",
    y = "PACF"
)

```

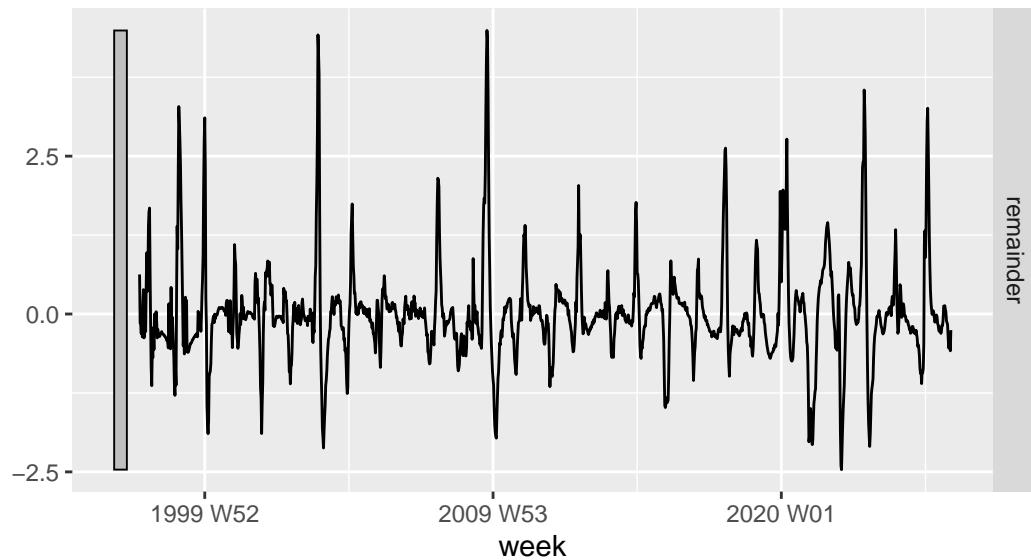
## Partial Autocorrelation Function of Weighted ILI (%)

PACF highlights autoregressive behavior and seasonal effects



```
# Check for outliers via STL remainder
ili_ts |>
  model(STL(weighted_ilis)) |>
  components() |>
  autoplot(remainder) +
  labs(title = "Outlier Inspection via STL Remainder")
```

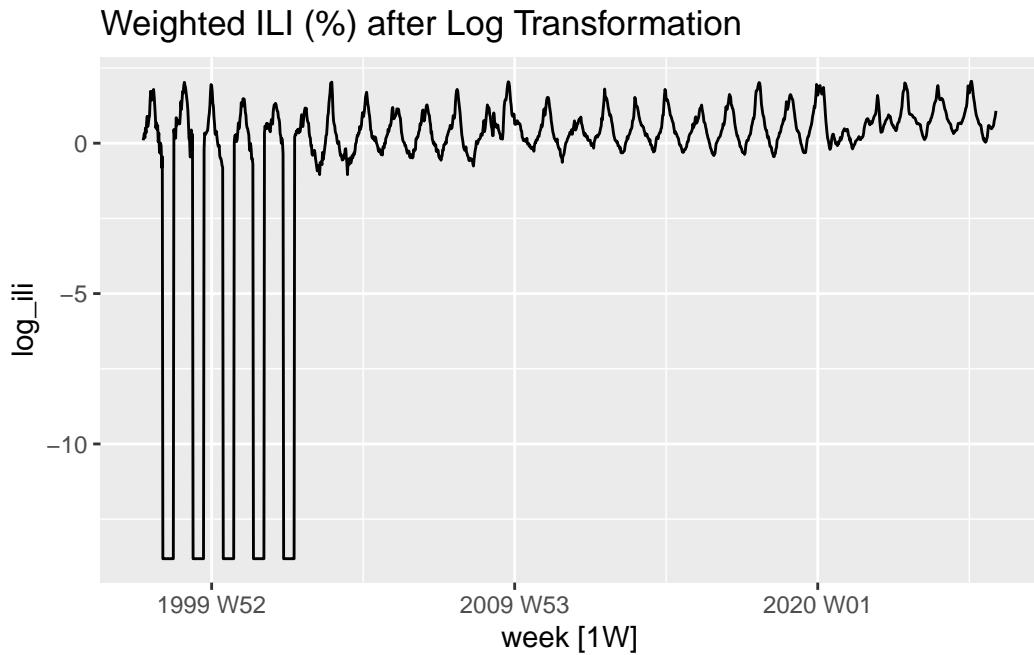
## Outlier Inspection via STL Remainder



## Pre-Processing and Data Preparation

```
# Log transformation
ili_ts <- ili_ts |>
  mutate(log_ili = log(weighted_ili + 1e-6))

ili_ts |> autoplot(log_ili) + ggtitle("Weighted ILI (%) after Log Transformation")
```



## Train/Test Split

```
# Train/test split
train <- ili_ts |> filter(week < yearweek("2020 W01"))
test  <- ili_ts |> filter(week >= yearweek("2020 W01"))
```

## Modeling

TSLM

```
fit_tslm <- train |>
  model(
    tslm = TSLM(weighted_ili ~ trend() + season())
  )

report(fit_tslm)
```

Series: weighted\_ili  
Model: TSLM

Residuals:

	Min	1Q	Median	3Q	Max
	-2.15990	-0.44007	-0.07707	0.19706	5.99450

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.616e+00	2.031e-01	7.956	4.35e-15 ***
trend()	6.372e-04	8.277e-05	7.698	3.05e-14 ***
season()year2	1.071e-01	2.790e-01	0.384	0.701280
season()year3	3.169e-01	2.790e-01	1.136	0.256311
season()year4	6.483e-01	2.790e-01	2.323	0.020336 *
season()year5	1.155e+00	2.790e-01	4.140	3.73e-05 ***
season()year6	1.258e+00	2.790e-01	4.509	7.22e-06 ***
season()year7	1.413e+00	2.790e-01	5.065	4.78e-07 ***
season()year8	1.485e+00	2.790e-01	5.324	1.23e-07 ***
season()year9	1.343e+00	2.790e-01	4.813	1.69e-06 ***
season()year10	1.456e+00	2.822e-01	5.160	2.93e-07 ***
season()year11	1.606e+00	2.822e-01	5.691	1.62e-08 ***
season()year12	1.622e+00	2.822e-01	5.750	1.15e-08 ***
season()year13	1.637e+00	2.822e-01	5.800	8.65e-09 ***
season()year14	1.409e+00	2.822e-01	4.993	6.89e-07 ***
season()year15	1.148e+00	2.822e-01	4.067	5.10e-05 ***
season()year16	8.503e-01	2.822e-01	3.013	0.002642 **
season()year17	5.406e-01	2.822e-01	1.916	0.055618 .
season()year18	2.827e-01	2.822e-01	1.002	0.316552
season()year19	-3.040e-03	2.822e-01	-0.011	0.991406
season()year20	-2.687e-01	2.822e-01	-0.952	0.341090
season()year21	-4.485e-01	2.822e-01	-1.590	0.112230
season()year22	-6.235e-01	2.822e-01	-2.210	0.027326 *
season()year23	-7.140e-01	2.822e-01	-2.530	0.011530 *
season()year24	-7.863e-01	2.822e-01	-2.787	0.005419 **
season()year25	-8.356e-01	2.822e-01	-2.961	0.003128 **
season()year26	-1.051e+00	2.822e-01	-3.724	0.000206 ***
season()year27	-1.093e+00	2.822e-01	-3.874	0.000113 ***
season()year28	-1.094e+00	2.822e-01	-3.876	0.000112 ***
season()year29	-1.137e+00	2.822e-01	-4.030	5.97e-05 ***
season()year30	-1.202e+00	2.822e-01	-4.259	2.22e-05 ***
season()year31	-1.250e+00	2.822e-01	-4.429	1.04e-05 ***
season()year32	-1.303e+00	2.822e-01	-4.617	4.36e-06 ***
season()year33	-1.324e+00	2.822e-01	-4.692	3.04e-06 ***
season()year34	-1.342e+00	2.822e-01	-4.756	2.24e-06 ***
season()year35	-1.387e+00	2.822e-01	-4.914	1.03e-06 ***

```

season()year36 -1.419e+00 2.822e-01 -5.031 5.70e-07 ***
season()year37 -1.427e+00 2.822e-01 -5.058 4.97e-07 ***
season()year38 -1.437e+00 2.822e-01 -5.093 4.13e-07 ***
season()year39 -1.435e+00 2.822e-01 -5.087 4.26e-07 ***
season()year40 -1.414e+00 2.822e-01 -5.010 6.33e-07 ***
season()year41 -1.348e+00 2.822e-01 -4.776 2.03e-06 ***
season()year42 -1.236e+00 2.822e-01 -4.381 1.29e-05 ***
season()year43 -1.155e+00 2.822e-01 -4.092 4.59e-05 ***
season()year44 -1.062e+00 2.790e-01 -3.805 0.000149 ***
season()year45 -7.582e-01 2.790e-01 -2.717 0.006682 **
season()year46 -6.157e-01 2.790e-01 -2.207 0.027530 *
season()year47 -5.496e-01 2.790e-01 -1.970 0.049125 *
season()year48 -4.106e-01 2.790e-01 -1.472 0.141431
season()year49 -2.967e-01 2.790e-01 -1.063 0.287871
season()year50 -2.307e-01 2.790e-01 -0.827 0.408575
season()year51 -1.402e-01 2.790e-01 -0.502 0.615526
season()year52 -9.887e-02 2.790e-01 -0.354 0.723127
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

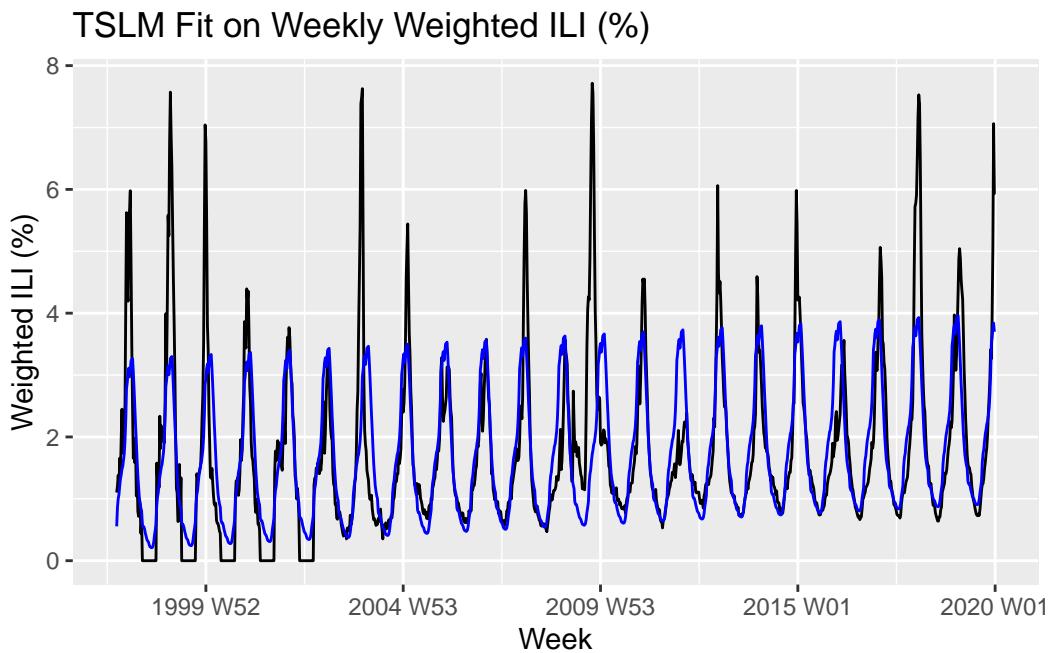
Residual standard error: 0.9462 on 1109 degrees of freedom  
 Multiple R-squared: 0.565, Adjusted R-squared: 0.5446  
 F-statistic: 27.7 on 52 and 1109 DF, p-value: < 2.22e-16

```

tslm_aug <- fit_tslm |>
augment()

autoplot(train, weighted_ili) +
autolayer(tslm_aug, .fitted, color = "blue") +
labs(
title = "TSLM Fit on Weekly Weighted ILI (%)",
x = "Week",
y = "Weighted ILI (%)"
)

```



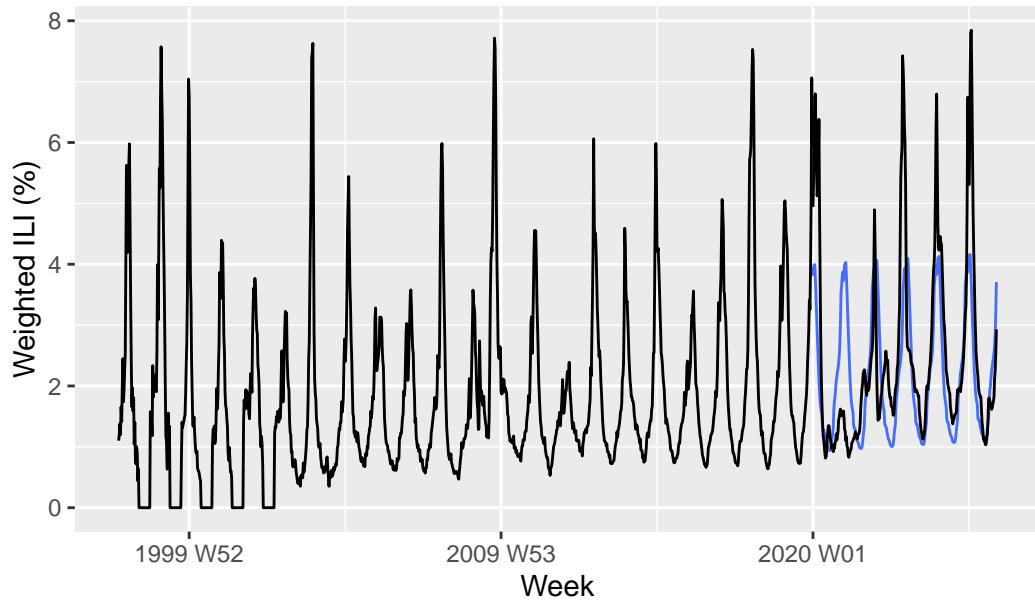
```

fc_tslm <- fit_tslm |>
forecast(h = nrow(test))

fc_tslm |>
autoplot(train, level = NULL) +
autolayer(test, weighted_ilis, color = "black") +
labs(
  title = "TSLM Forecast vs Actual (Original Scale)",
  x = "Week",
  y = "Weighted ILI (%)"
)

```

### TSLM Forecast vs Actual (Original Scale)



```
accuracy_tslm <- fc_tslm |>
accuracy(test)
```

```
accuracy_tslm
```

```
# A tibble: 1 x 10
.model .type    ME   RMSE    MAE    MPE    MAPE    MASE RMSSE   ACF1
<chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 tslm  Test  0.287  1.25  0.877 -3.63  38.0   NaN    NaN  0.951
```

### ARIMA Modeling

```
# Fit model on training data
fit_arima <- train |>
  model(
    arima = ARIMA(log_ili)
  )

# Review model output
report(fit_arima)
```

```
Series: log_ili
```

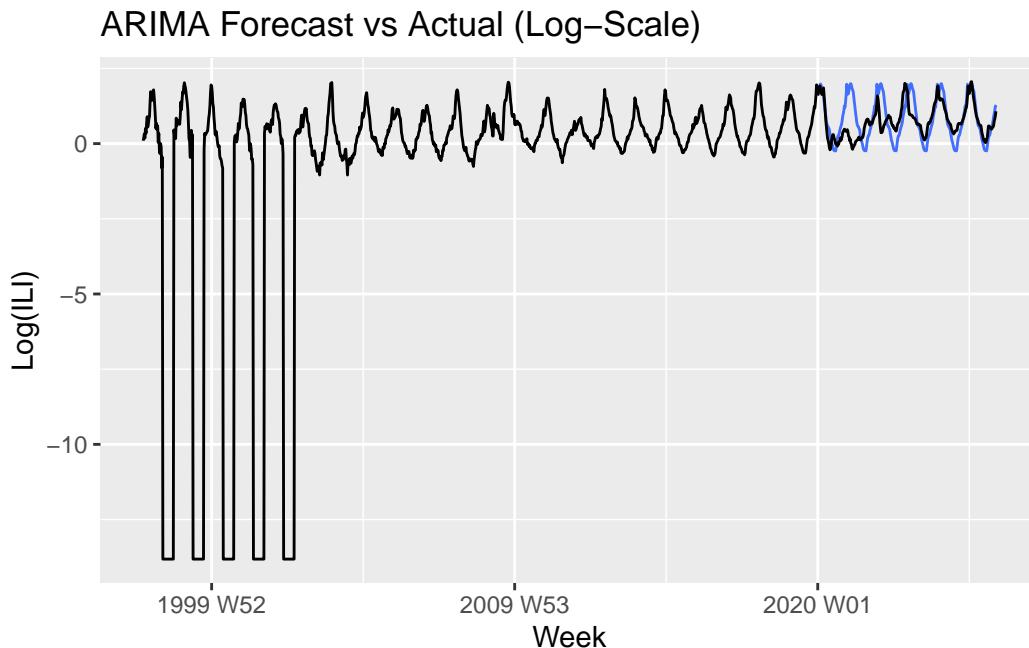
```
Model: ARIMA(2,0,0)(0,1,0) [52]
```

```
Coefficients:
```

	ar1	ar2
0.9645	-0.0254	
s.e.	0.0368	0.0368

```
sigma^2 estimated as 0.5372: log likelihood=-1230.23  
AIC=2466.46 AICc=2466.48 BIC=2481.49
```

```
# Forecast with Test  
fc_arima <- fit_arima |> forecast(h= nrow(test))  
  
# Plot forecast versus actuals  
fc_arima |>  
  autoplot(train, level = NULL) +  
  autolayer(test, log_ili, color = "black") +  
  ggtitle("ARIMA Forecast vs Actual (Log-Scale)") +  
  xlab("Week") + ylab("Log(ILS)")
```



```
# Accuracy metrics  
accuracy_arima <- fc_arima |>  
accuracy(test)
```

```
accuracy_arima
```

```
# A tibble: 1 x 10
  .model .type      ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
  <chr>  <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 arima  Test  -0.0790 0.595 0.439 -136.  418.    NaN    NaN  0.979
```

## ETS Modeling

```
# ETS models: manual non-seasonal ETS(A,A,N) and auto ETS with no seasonality
fit_ets <- train |>
  model(
    ets_manual = ETS(weighted_ilis ~ error("A") + trend("A") + season("N")), # ETS(A,A,N)
    ets_auto   = ETS(weighted_ilis ~ season("N"))                           # auto ETS, non
  )

report(fit_ets)
```

```
# A tibble: 2 x 9
  .model      sigma2 log_lik     AIC    AICc     BIC     MSE    AMSE    MAE
  <chr>        <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 ets_manual  0.152 -3003. 6016. 6016. 6041. 0.151 0.535 0.220
2 ets_auto    0.135 -2937. 5885. 5886. 5916. 0.135 0.439 0.207
```

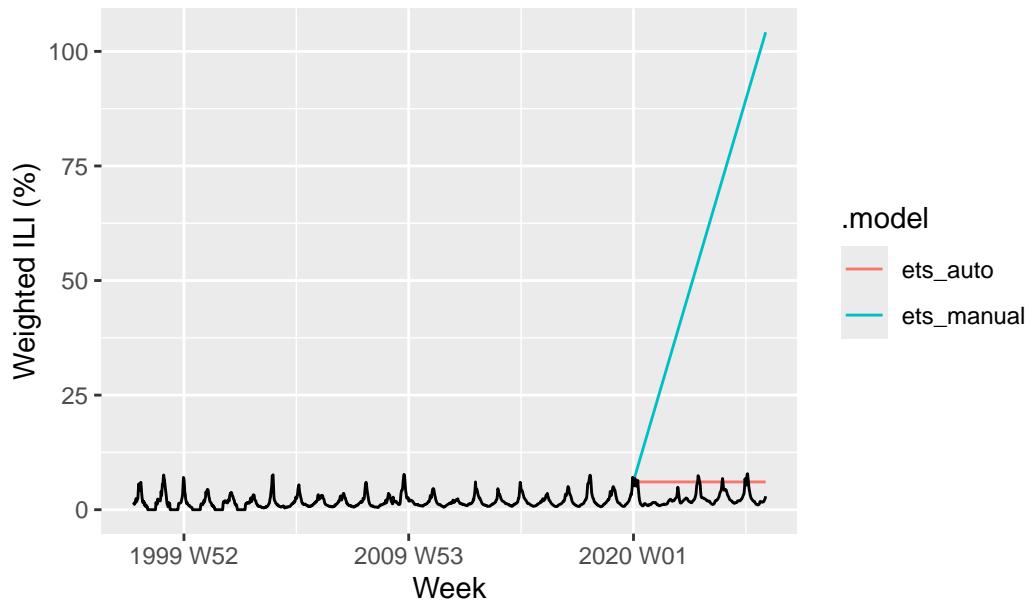
```
# Forecast the same horizon as the test set

fc_ets <- fit_ets |>
  forecast(h = nrow(test))

# Plot ETS forecasts vs actual values on the original scale

fc_ets |>
  autoplot(train, level = NULL) +
  autolayer(test, weighted_ilis, color = "black") +
  labs(
    title = "ETS Forecasts vs Actual (Original Scale)",
    x = "Week",
    y = "Weighted ILI (%)"
  )
```

## ETS Forecasts vs Actual (Original Scale)



```
# Accuracy metrics for ETS models on the test set
```

```
accuracy_ets <- fc_ets |>
accuracy(test)
```

```
accuracy_ets
```

```
# A tibble: 2 x 10
```

.model	.type	ME	RMSE	MAE	MPE	MAPE	MASE	RMSSE	ACF1
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 ets_auto	Test	-3.58	3.91	3.67	-226.	228.	NaN	NaN	0.968
2 ets_manual	Test	-52.7	59.8	52.7	-2658.	2658.	NaN	NaN	0.990

```
# Manual vs Auto ETS forecasts plotted over the full series
```

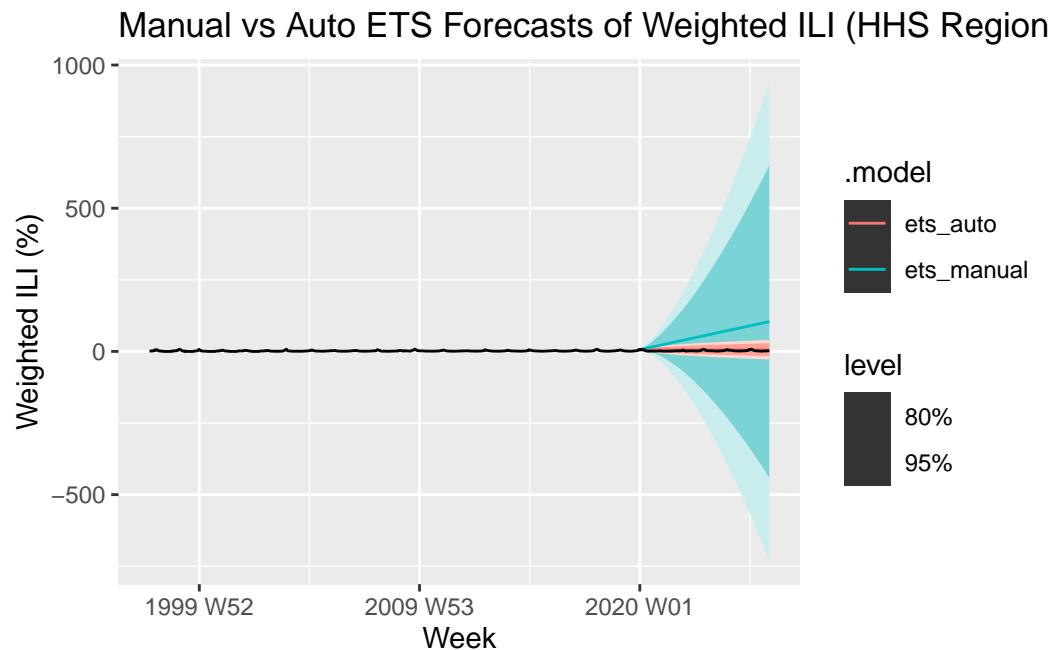
```
fc_ets_full <- fit_ets |>
forecast(h = nrow(test))
```

```
fc_ets_full |>
autoplot(ili_ts) +
labs(
title = "Manual vs Auto ETS Forecasts of Weighted ILI (HHS Region 4)",
```

```

x = "Week",
y = "Weighted ILI (%)"
)

```



```

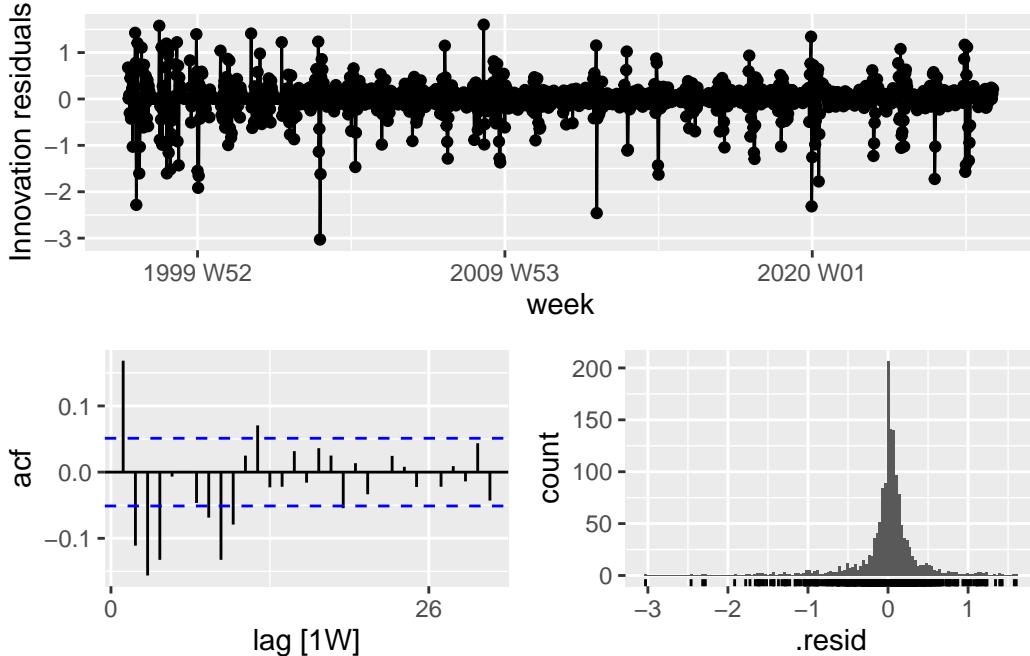
# Fit a manual ETS(A,A,N) model on the full series for diagnostics

fit_ets_full <- ili_ts |>
model(
ets_manual = ETS(weighted_ilis ~ error("A") + trend("A") + season("N"))
)

# Residual diagnostics plot (like gg_tsresiduals in the ETS document)

fit_ets_full |>
gg_tsresiduals()

```



```
# Ljung-Box test on residuals (check for remaining autocorrelation)

ets_ljung_box <- fit_ets_full |>
augment() |>
features(.innov, ljung_box, lag = 24)

ets_ljung_box
```

```
# A tibble: 1 x 3
.model      lb_stat lb_pvalue
<chr>        <dbl>     <dbl>
1 ets_manual   189.       0
```

```
# Fit ETS models on the full series
fit_ets_all_full <- ili_ts |>
model(
  ets_manual = ETS(weighted_ilis ~ error("A") + trend("A") + season("N")), # ETS(A,A,N)
  ets_auto   = ETS(weighted_ilis ~ season("N")), # auto ETS, no
  ets_damped = ETS(weighted_ilis ~ error("A") + trend("Ad") + season("N")) # damped trend
)

# Forecast ahead (e.g., next 52 weeks)
```

```

fc_ets_all_full <- fit_ets_all_full |>
  forecast(h = 52)

# Plot like the original ETS figure
fc_ets_all_full |>
  autoplot(ili_ts, level = c(80, 95)) +
  coord_cartesian(ylim = c(0, 10)) + # adjust if your %wILI scale is different
  labs(
    title = "Manual vs Auto vs Damped ETS Forecasts of %wILI (HHS Region 4)",
    x = "Year-Week",
    y = "%wILI"
  )

```

