Citi Bike in NYC

Forecasting Univariate Time Series with SARIMA and Exponential Smoothing

Elise Hage (SNR: i64282956),
Timothy Hanson (SNR: i6286194) and
Selina Blom (SNR: i6327002)

(ETS)

Content

RIDE EXPERIENCE

Get to know Citi Bike

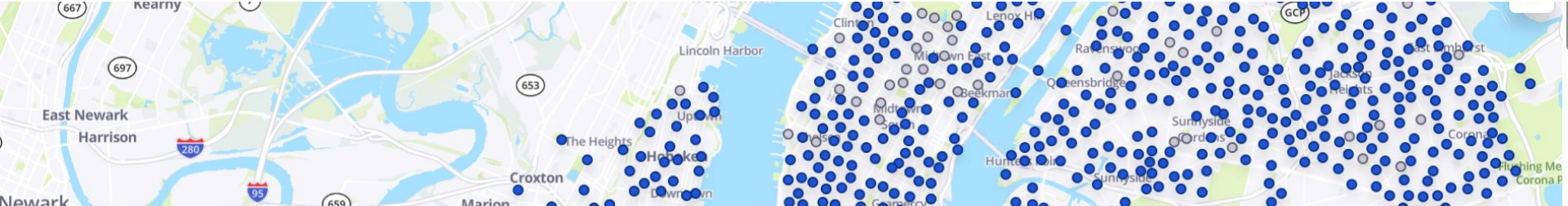
Download the app

Expanding access to bikes

Citi Bike is the US nation's largest bikeshare program, with 25,000 bikes and over 1,500 stations across Manhattan, Brooklyn, Queens, the Bronx, Jersey City, and Hoboken.

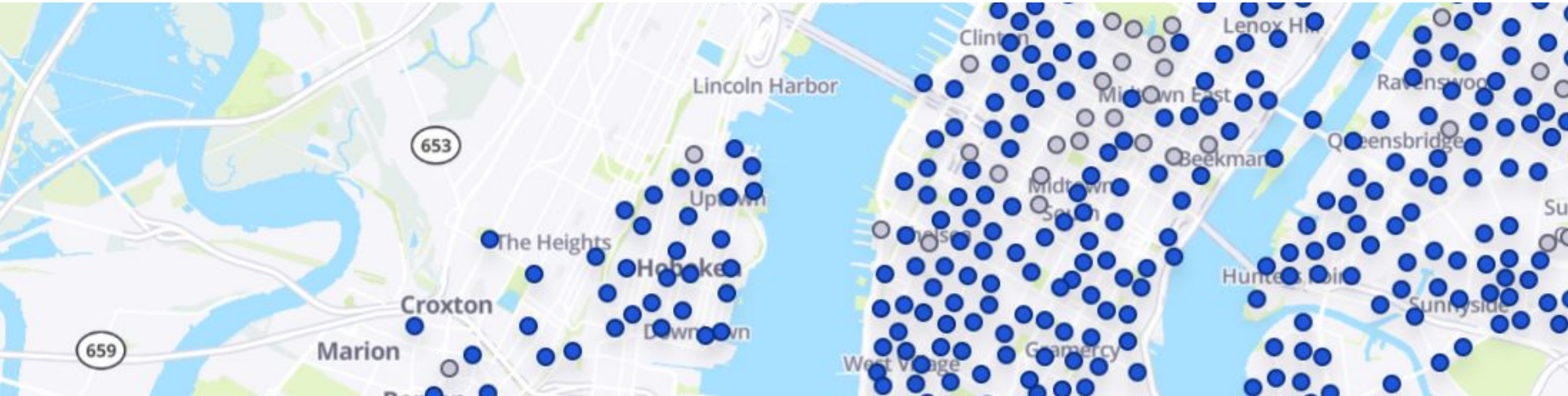
Variables:

- year: numeric, indicating the year
- month: numeric, indicating the month of the year (1-12)
- day: numeric, indicating the day of the month (1-31)
- hour: numeric, indicating the hour of the day (0-23)
- wkday: numeric, indicating the day of the week (1 for Monday until 7 for Sunday)
- demand: numeric, the number of bikes picked up at the location



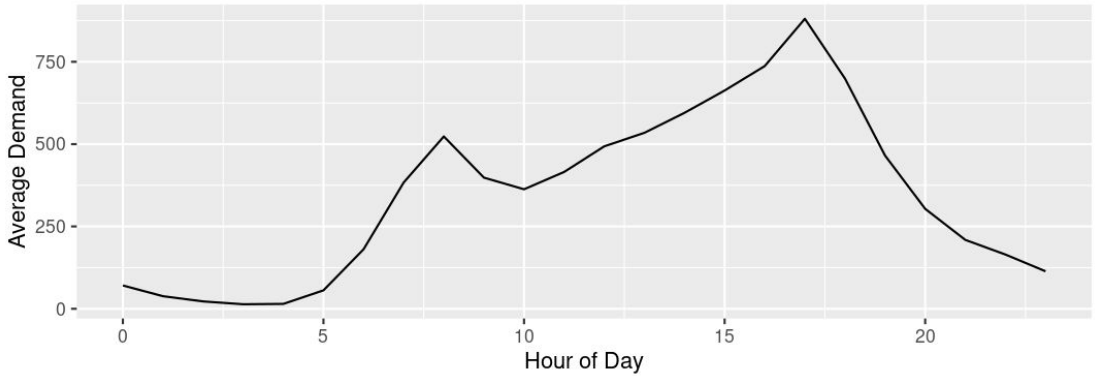
NOTE: The trip data can only be seen as a proxy for the true demand

- Demand is 'observed trips'.
- Realized outcome.

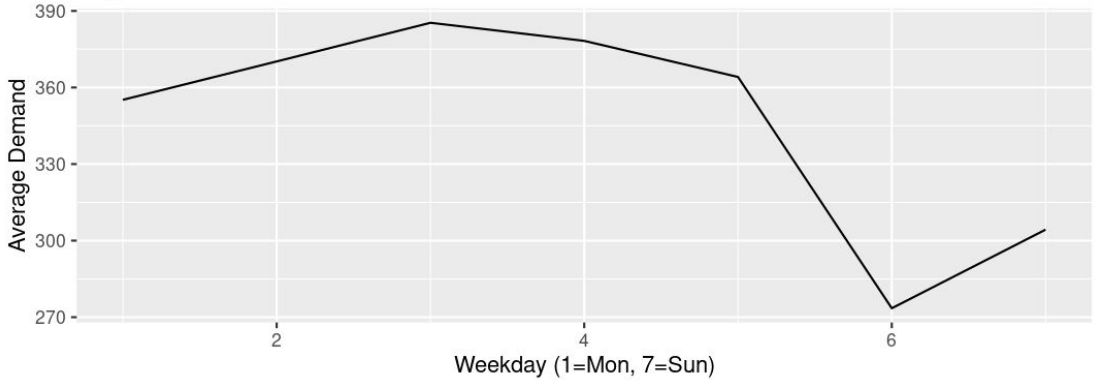


Data Exploration

Average Hourly Demand Pattern



Day-of-Week Pattern



Shared micromobility supports riders ...

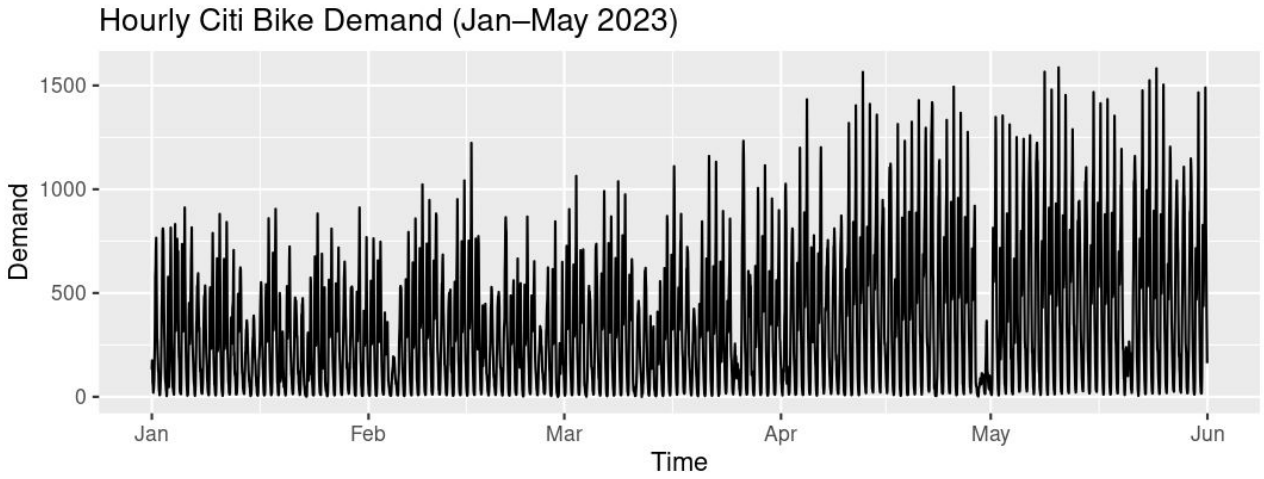
46%
to get **to or from work**

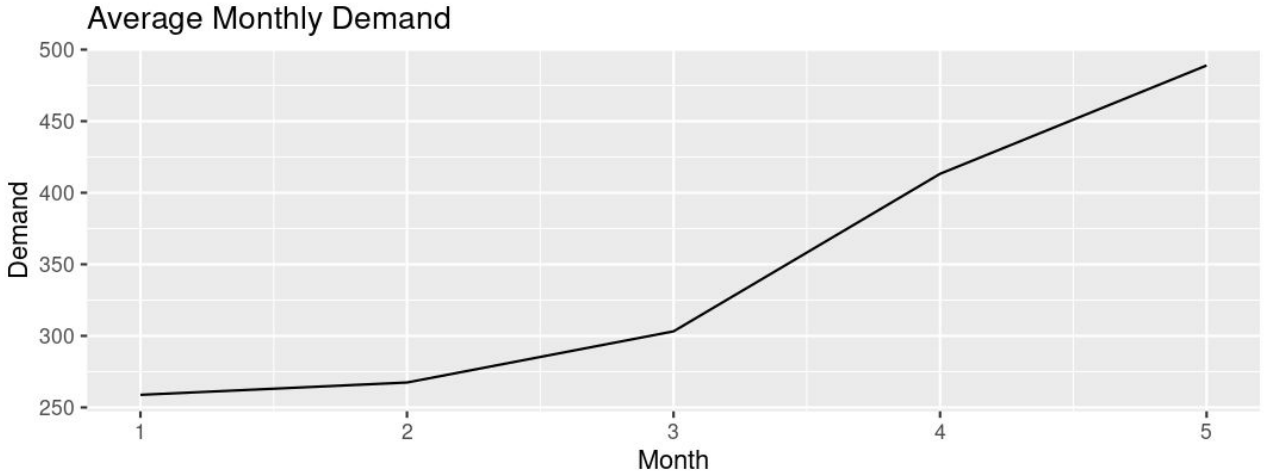
62%
to run **errands**

18%
to access **healthcare services**

47%
for **commercial recreation/ entertainment**

Demand of Citi Bike is higher in some months:





Stationarity

- Constant mean/variance overtime
- No long-term trends.

```
Value of test-statistic is: -15.0987

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Augmented Dickey-Fuller (ADF) Test

Null hypothesis: The series has a unit root.
Alternative hypothesis: The series does **not** have a unit root.

Problems:

- Low power;
- Checks one type of non-stationarity;
- Mislead you if series is trend-stationary.

Stationarity

```
########################
# KPSS Unit Root Test #
########################

Test is of type: mu with 9 lags.

Value of test-statistic is: 5.5981

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

KPSS test

The **KPSS** (Kwiatkowski–Phillips–Schmidt–Shin) test checks almost the opposite of ADF:

Null Hypothesis: series is (trend) stationary.
Alternative Hypothesis: series is non-stationary.

This makes the KPSS test useful:

- Detect trend-stationary behavior.
- Second perspective.

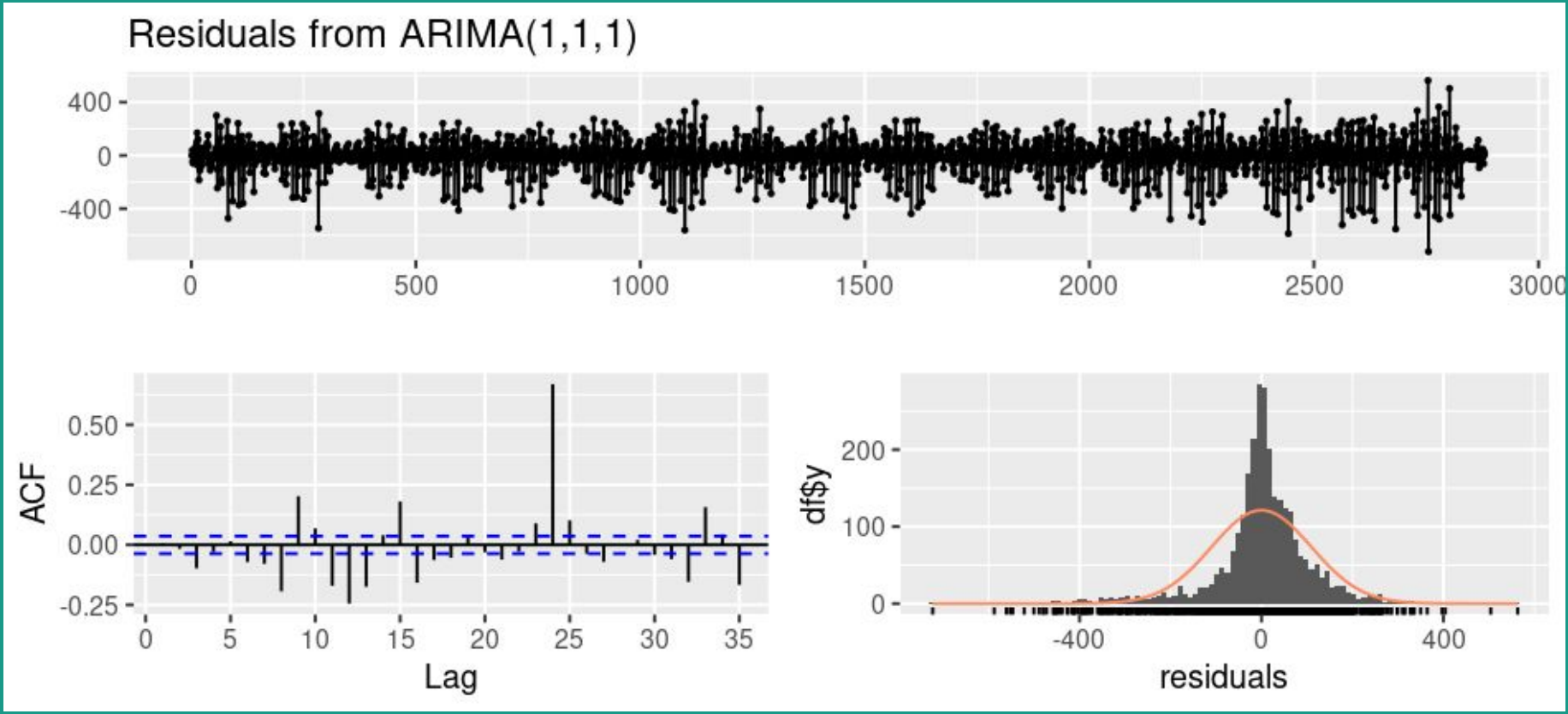
Output: d=1

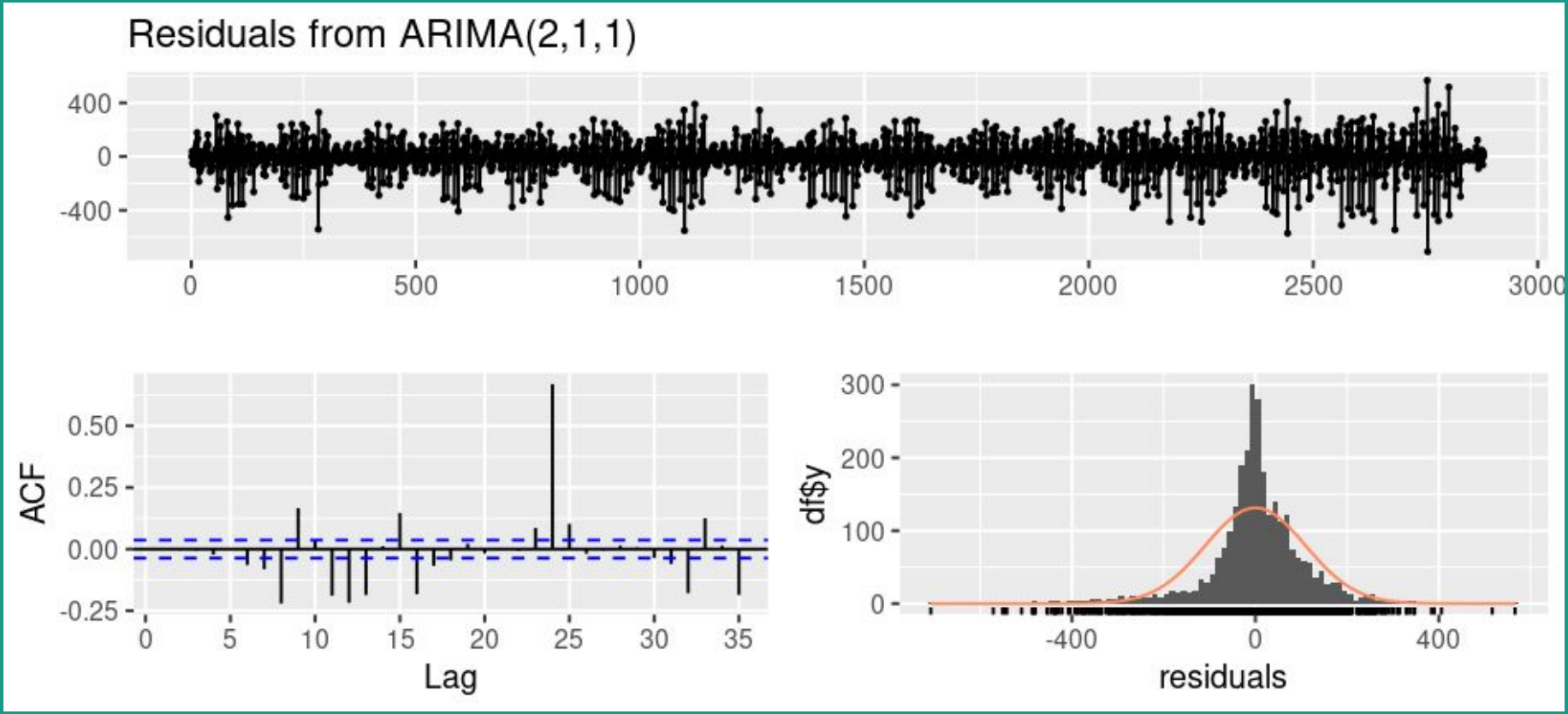
ARIMA comparisons

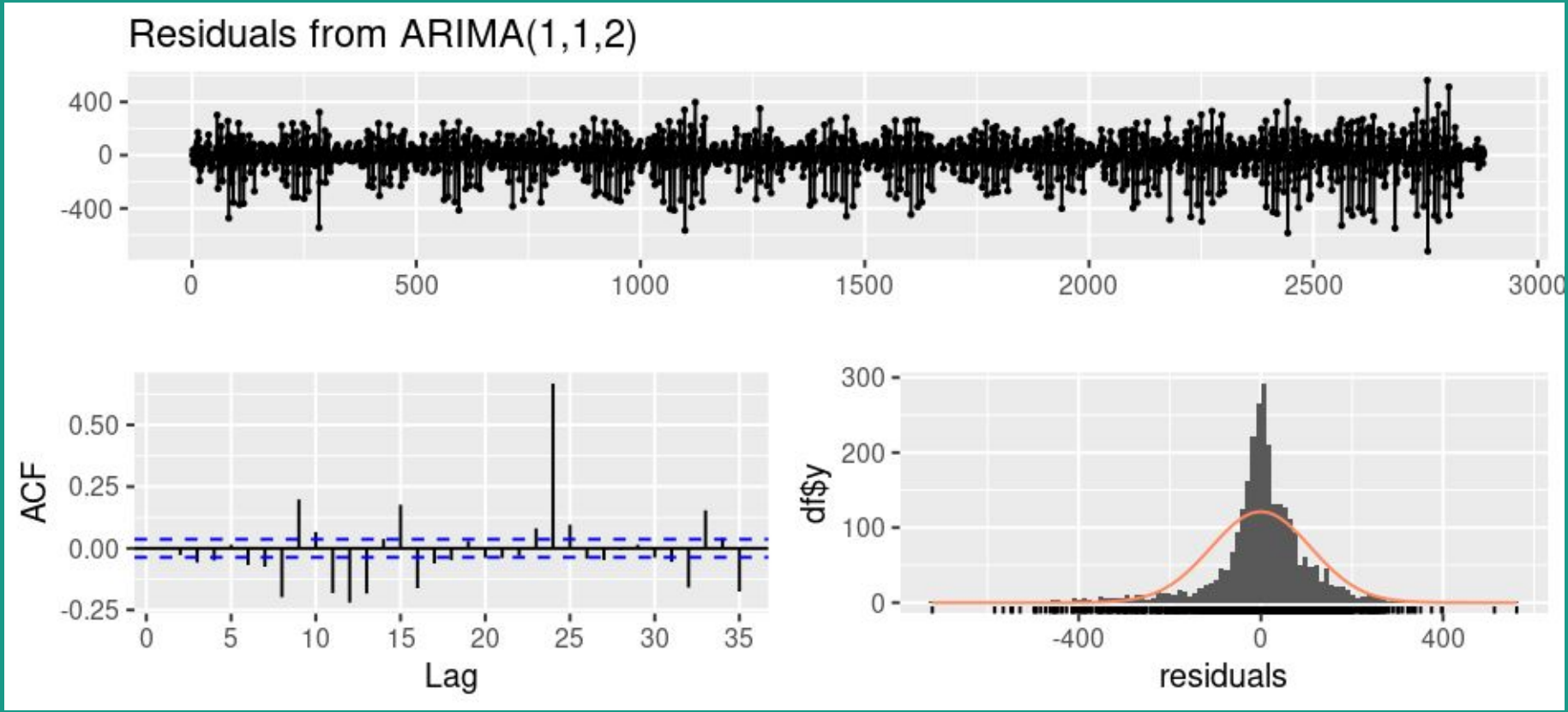
We compare 2 criteria: AIC and BIC:

- Lower values of AIC/BIC are better.

Model	AIC	BIC
ARIMA_111	35233.04	35250.94
ARIMA_211	35204.13	35227.99
ARIMA_112	35224.20	35248.06
Automatic ARIMA	35202.4	35220.29

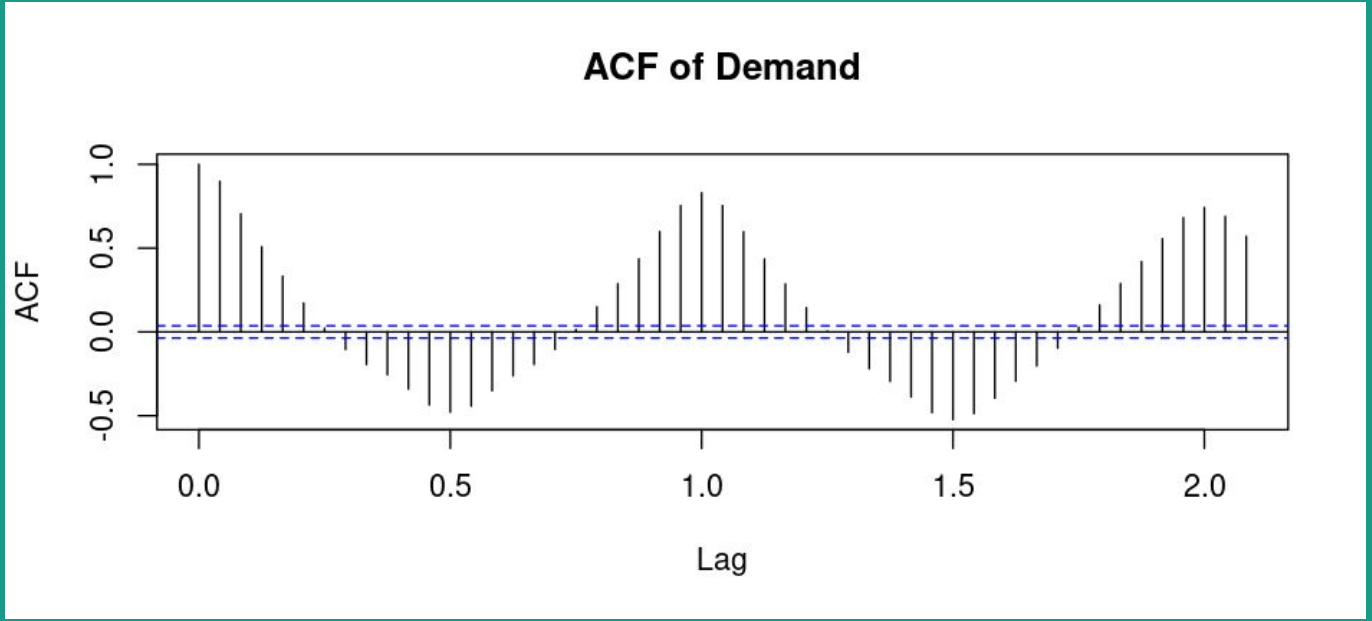

Residuals from ARIMA(1,1,1)
400
0
-400
0
500
1000
1500
2000
2500
3000
ACF
0.50
0.25
0.00
-0.25
0
5
10
15
20
25
30
35
Lag
df$y
200
100
0
-400
0
400
residuals


Residuals from ARIMA(2,1,1)
400
0
-400
0
500
1000
1500
2000
2500
3000
ACF
0.50
0.25
0.00
-0.25
0
5
10
15
20
25
30
35
Lag
df$y
300
200
100
0
-400
0
400
residuals


Residuals from ARIMA(1,1,2)
400
0
-400
0
500
1000
1500
2000
2500
3000
ACF
0.50
0.25
0.00
-0.25
0
5
10
15
20
25
30
35
Lag
df$y
300
200
100
0
-400
0
400
residuals

Seasonality

Daily seasonality.
Lags are in days.



Stationarity

Remember:

- ADF says stationary
- KPSS says non-stationary

-> Seasonal unit roots

Output:
Seasonal differencing D=0

Canova Hansen (CH) Test

Null Hypothesis: No seasonal unit roots
Alternative Hypothesis: Seasonal unit roots

If the test indicates seasonal unit roots → Apply seasonal differencing (lag = seasonal frequency).

SARIMA

01

SARIMA

01

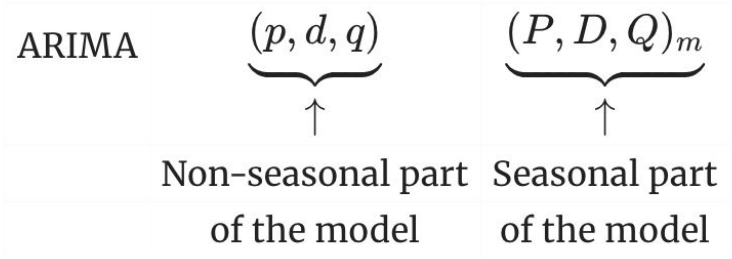
Why Seasonal ARIMA?

- Extends non-seasonal ARIMA to model seasonal patterns in time series
- Captures repeated cyclical behaviour.

SARIMA

01

- **Non-seasonal part:** (p,d,q)
- **Seasonal part:** (P,D,Q)_m
- **m:** seasonal period ,P:number of seasonal AR terms,D:number of seasonal differences,q:number of seasonal MA terms
- Lowercase = non-seasonal parameters
- Uppercase = seasonal parameters



Modelling Procedure

Same steps as non-seasonal ARIMA, but also:

- Select seasonal AR and MA terms
- Assess seasonal differencing needs

Select model with lowest AIC ->
ARIMA(2,0,0)(2,1,0)

Interesting: d=0 and D=1 !

SARIMA Comparisons

MANUAL & Auto

Automatic SARIMA:
ARIMA(2,0,0)(2,1,0)

D=1 -> removes 24 observation instead of 1

Model	df	AIC	BIC
SARIMA_112_101	6	33050.71	33086.50
SARIMA_211_101	6	33002.67	33038.46
Automatic SARIMA	5	33008.93	33038.72

Interpreting SARIMA

SARIMA(1,1,2)(1,0,1) - Effects on <u>current</u> change

AR1: Previous hour *change* of demand (d=1)

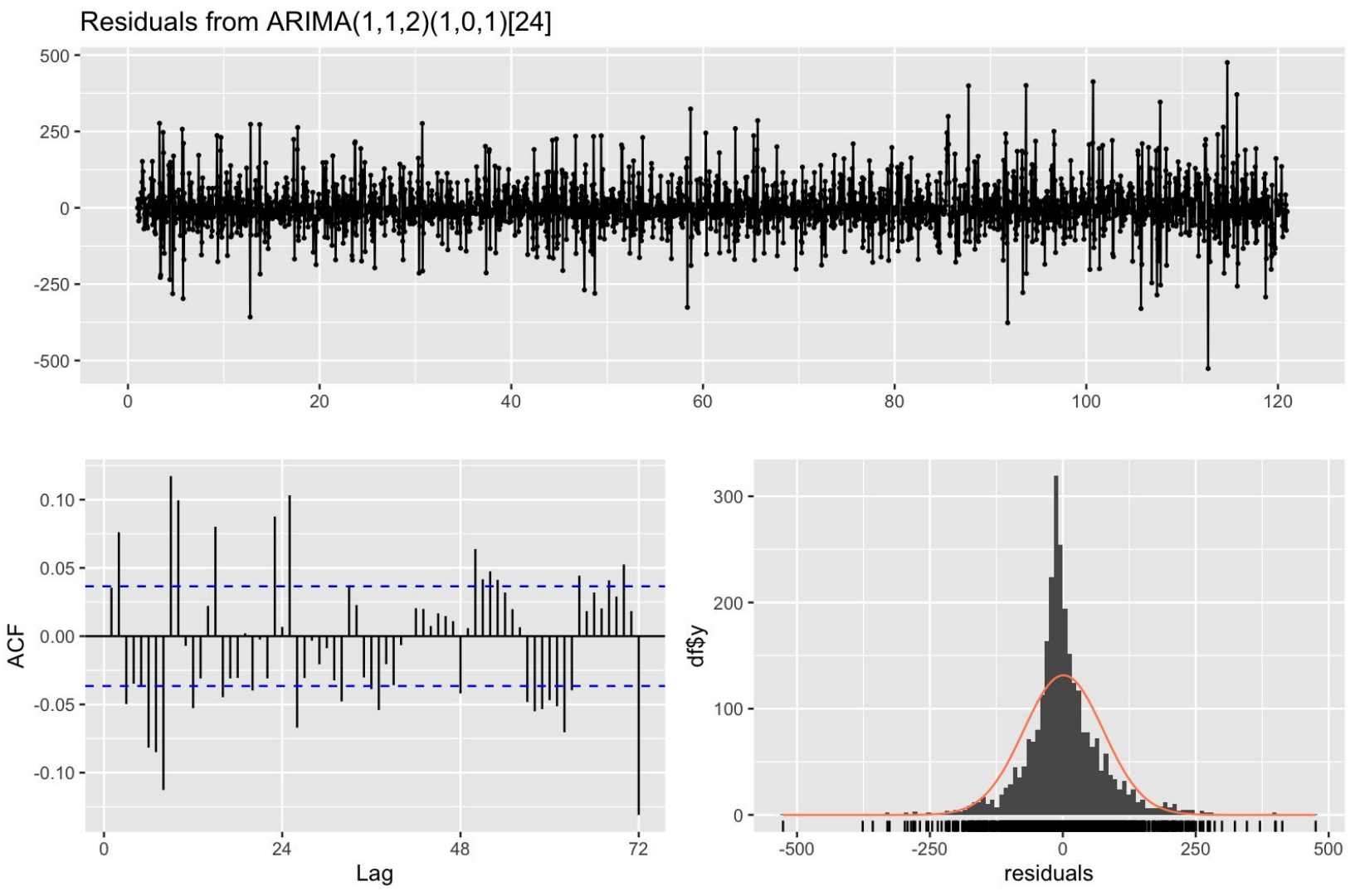
MA1: Shock in demand *change* previous hour

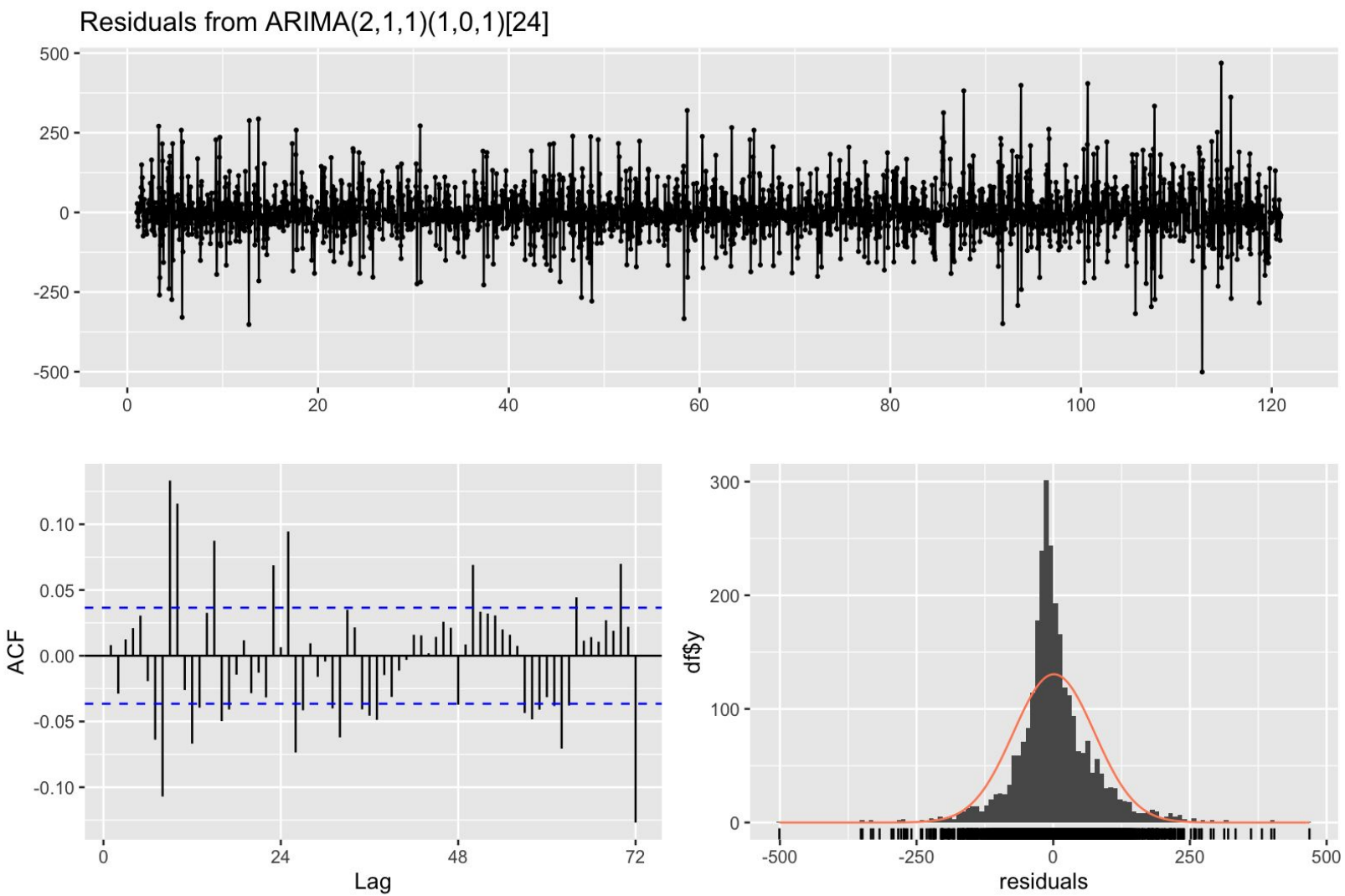
MA2: Shock in demand *change* two hours ago

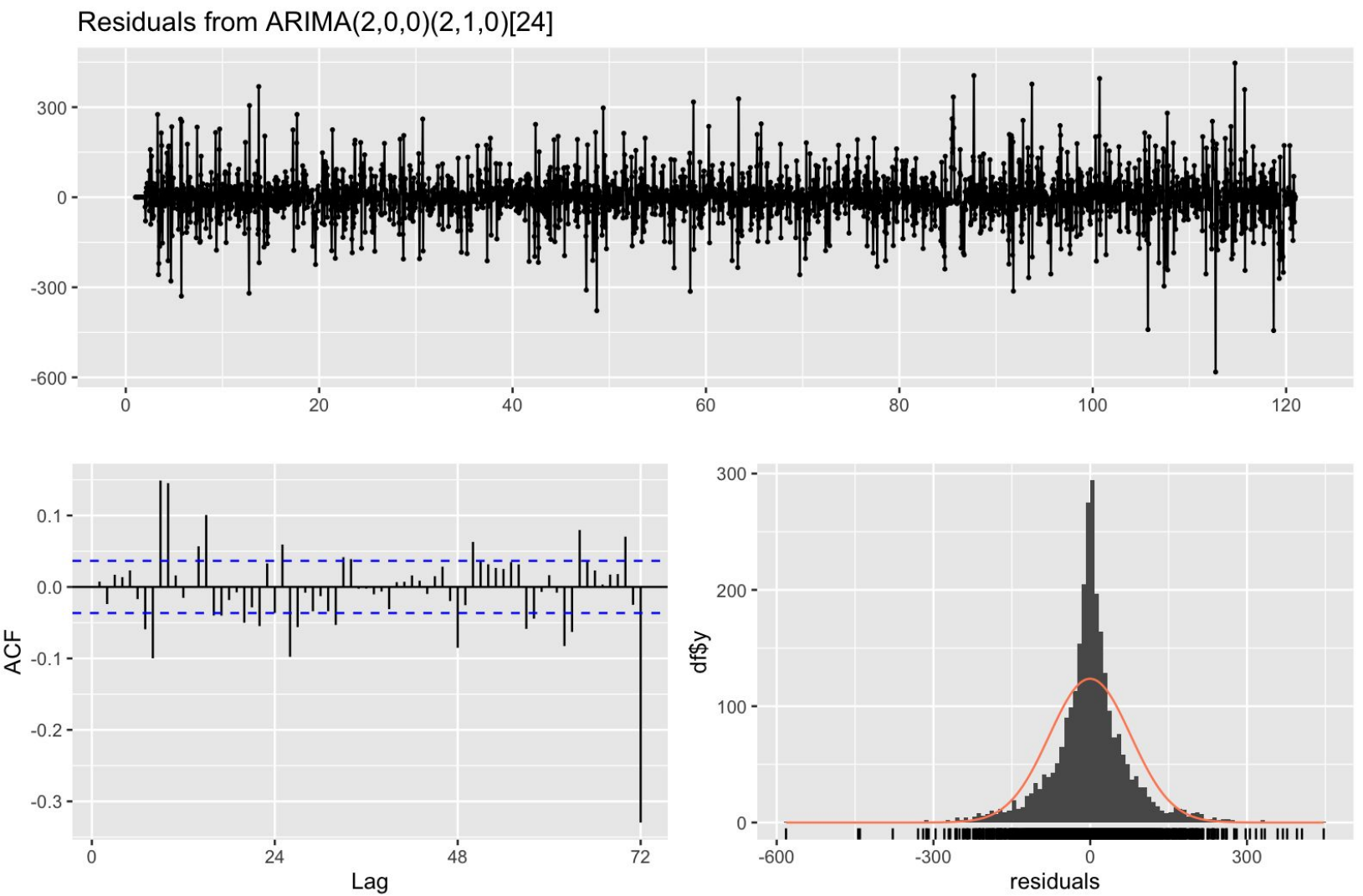
SAR1: Same hour, *change* in demand yesterday

SMA1: Shock at same hour yesterday

AR1	0.7717
MA1	-0.6053
MA2	-0.3939
SAR1	0.7646
SMA1	-0.0926


Residuals from ARIMA(1,1,2)(1,0,1)[24]
500
250
0
-250
-500
0
20
40
60
80
100
120
0.10
0.05
0.00
-0.05
-0.10
ACF
0
24
48
72
Lag
300
200
100
0
df$y
-500
-250
0
250
500
residuals


Residuals from ARIMA(2,1,1)(1,0,1)[24]
500
250
0
-250
-500
0
20
40
60
80
100
120
ACF
0.10
0.05
0.00
-0.05
-0.10
0
24
48
72
Lag
df$y
300
200
100
0
-500
-250
0
250
500
residuals


Residuals from ARIMA(2,0,0)(2,1,0)[24]
300
0
-300
-600
0
20
40
60
80
100
120
0.1
0.0
-0.1
-0.2
-0.3
ACF
0
24
48
72
Lag
300
200
100
0
df$y
-600
-300
0
300
residuals

Exponential Smoothing

02

Error	Additive, Multiplicative
Trend	None, Additive, Multiplicative, Additive damped, Multiplicative damped
Seasonal	None, Additive, Multiplicative

Exponential Smoothing

Results

$$y_t = f(\ell_{t-1}, b_{t-1}, s_{t-m}) + \varepsilon_t$$

- l: level that represent the baseline
- b: trend (can be damped)
- s: seasonality with pattern m
- e: error

Select model that minimizes AIC:

$$\mathrm{AIC} = -2\log(\hat{L}) + 2p,$$

Additive Error (No trend, no seasonality)

$$y_t = \ell_{t-1} + \varepsilon_t,$$
$$\ell_t = \ell_{t-1} + \alpha\varepsilon_t, \quad 0 < \alpha < 1.$$

Additive Error & Trend (no seasonality)

$$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t,$$
$$\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t,$$
$$b_t = b_{t-1} + \beta\,\varepsilon_t, \quad 0 < \beta < 1.$$

Additive Error & Seasonality (no trend)

$$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t,$$
$$\ell_t = \ell_{t-1} + \alpha\,\varepsilon_t,$$
$$s_t = s_{t-m} + \gamma\,\varepsilon_t, \quad 0 < \gamma < 1.$$

Comparison with SARIMA

1

Model Structure

ETS: pick from defined states, state space model

SARIMA: linear ARIMA model, potentially very large space

2

Stationarity

ETS: Stationary not required

SARIMA: Must be stationary

3

Non-linear

ETS: all components can be multiplicative -> can be a non-linear model

SARIMA: only linear

4

Results?

Hyndman (2001) speculates:

Small model space -> less estimation uncertainty -> often better forecasting

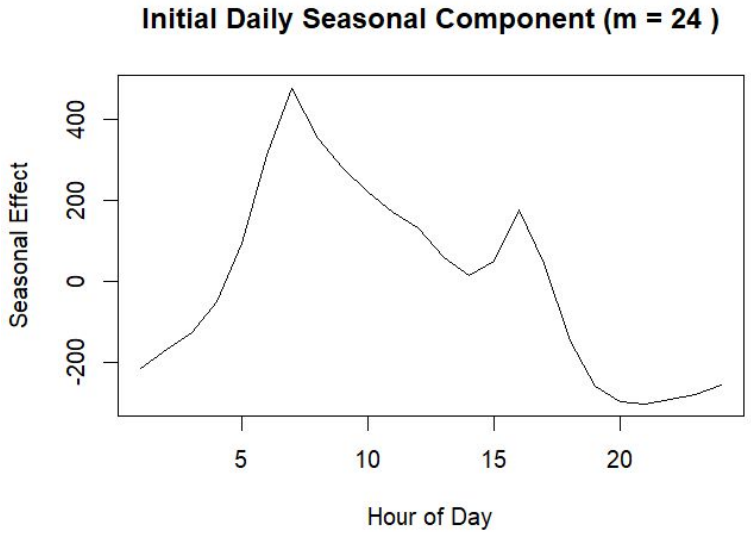
Exponential Smoothing

Selecting the Model

Additive Error & Seasonality (no trend)

$$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t,$$
$$\ell_t = \ell_{t-1} + \alpha\,\varepsilon_t,$$
$$s_t = s_{t-m} + \gamma\,\varepsilon_t, \quad 0 < \gamma < 1.$$

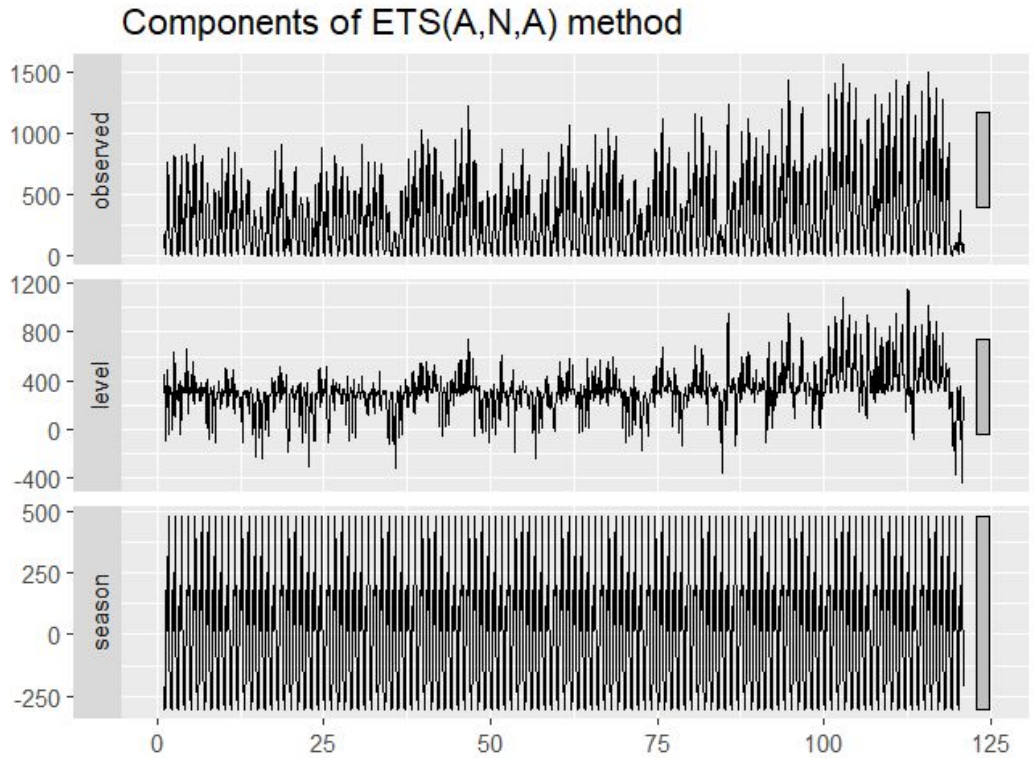
- $\alpha = 0.9999$, $l_0 = 286.8707$
- $\gamma = 0.0001$



Exponential Smoothing

Interpreting the Model

- Level updates aggressively based on error → more spikes, handles short-term noise
- Seasonal effect hardly updates → initial cycle is close to model seasonality, pattern



In-sample comparisons

We compare 3 criteria: AIC, BIC, LogLik:

- Lower values of AIC/BIC are better.
- Higher (less negative) values of LogLik is better.

Model	AIC	BIC	LogLik
ARIMA_111	35233.04	35250.94	-17613.52
ARIMA_211	35204.13	35227.99	-17598.06
ARIMA_112	35224.20	35248.06	-17608.10
SARIMA_112_101	33050.71	33086.50	-16519.35
SARIMA_211_101	33002.67	33038.46	-16495.33
SARIMA_auto	33008.93	33038.72	-16499.47
ETS	48315.51	48476.58	-24130.76

Conclusion: In-sample comparison

SARIMA_211_101 (but very close to SARIMA_auto) was best by AIC/BIC and Log-Lik

ETS fails to capture structure of TS

- Worst AIC/BIC

Rolling Window Forecast

- ➔ First iteration: January 1st - April 30th
- ➔ Second iteration: January 2nd - May 1st
- ➔ .
- ➔ .
- ➔ .
- ➔ Final iteration: February 1st - May 31st

Better reflects on model adaptability

Expanding Window Forecast

- ➔ First iteration: January 1st - April 30th
- ➔ Second iteration: January 1st - May 1st
- ➔ .
- ➔ .
- ➔ .
- ➔ Final iteration: January 1st - May 31st

Stabilize parameter estimates over time -> may reduce variance of predictions

Out of Sample Comparisons

Models Compared

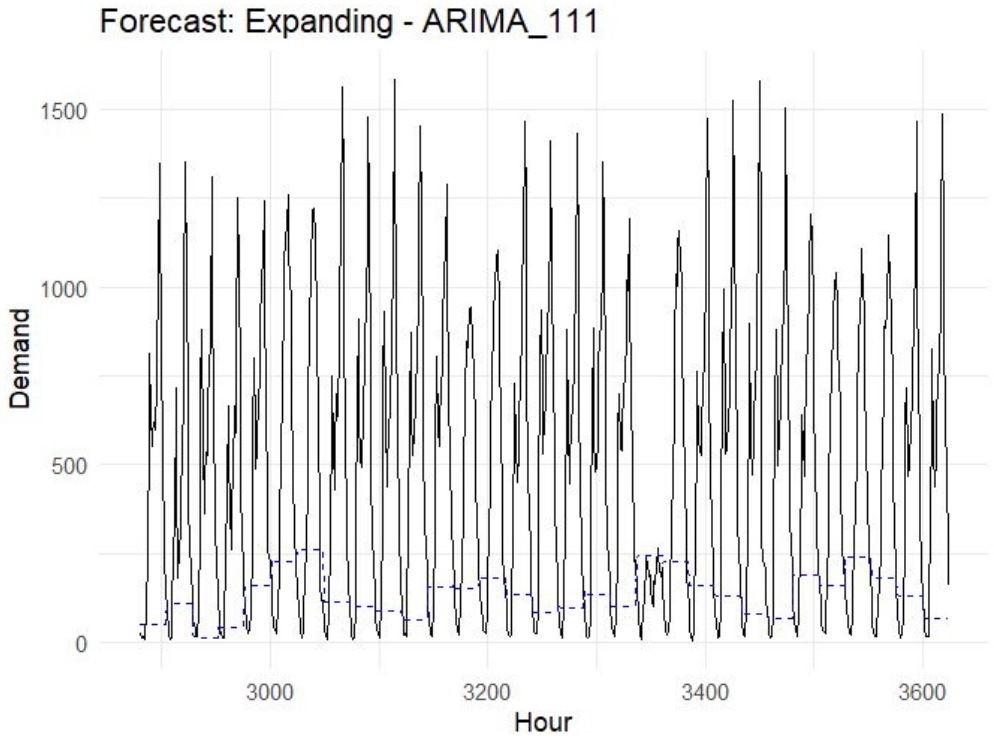
- ARIMA(1,1,1)
- ARIMA(2,1,1)
- ARIMA(1,1,2)
- ARIMA_AUTO
- SARIMA(1,1,2)(2,1,1)
- SARIMA(2,1,1)(1,0,1)
- SARIMA_AUTO
- Exponential smoothing

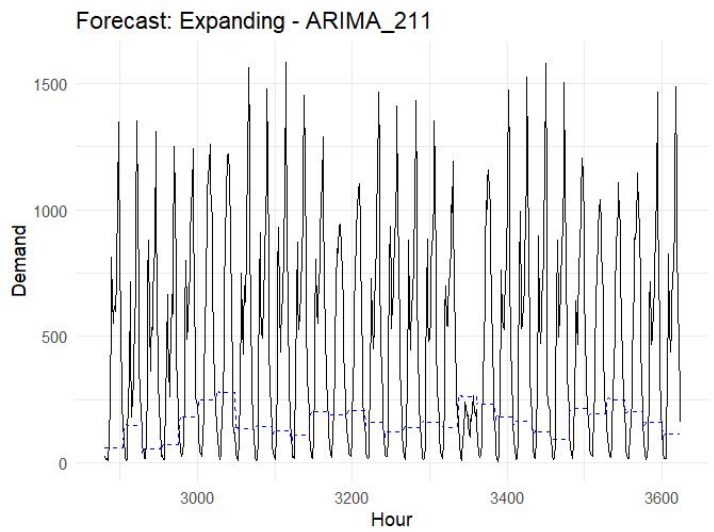
Window Forecasts

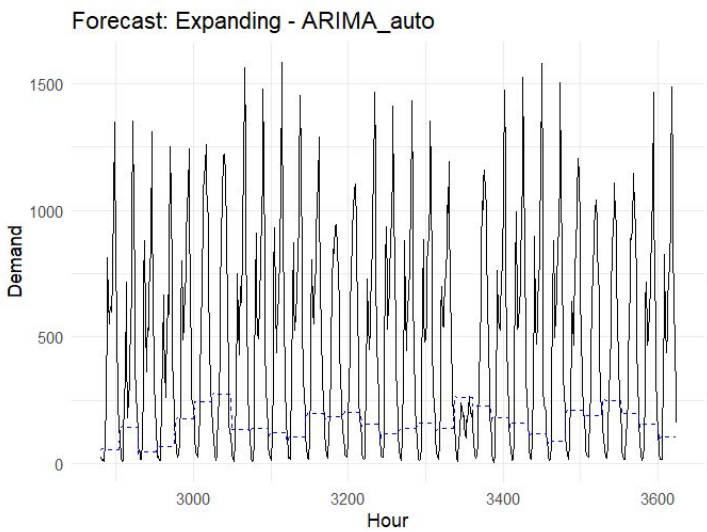
Blue = forecast

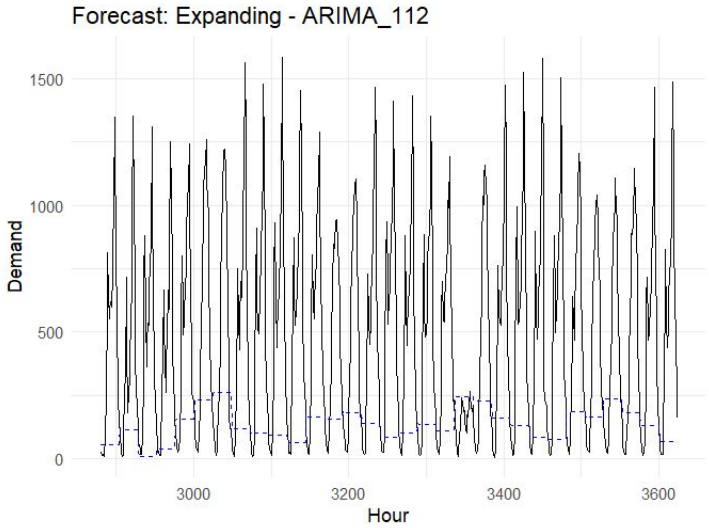
Black = actual

No seasonality:
Conservative estimates!



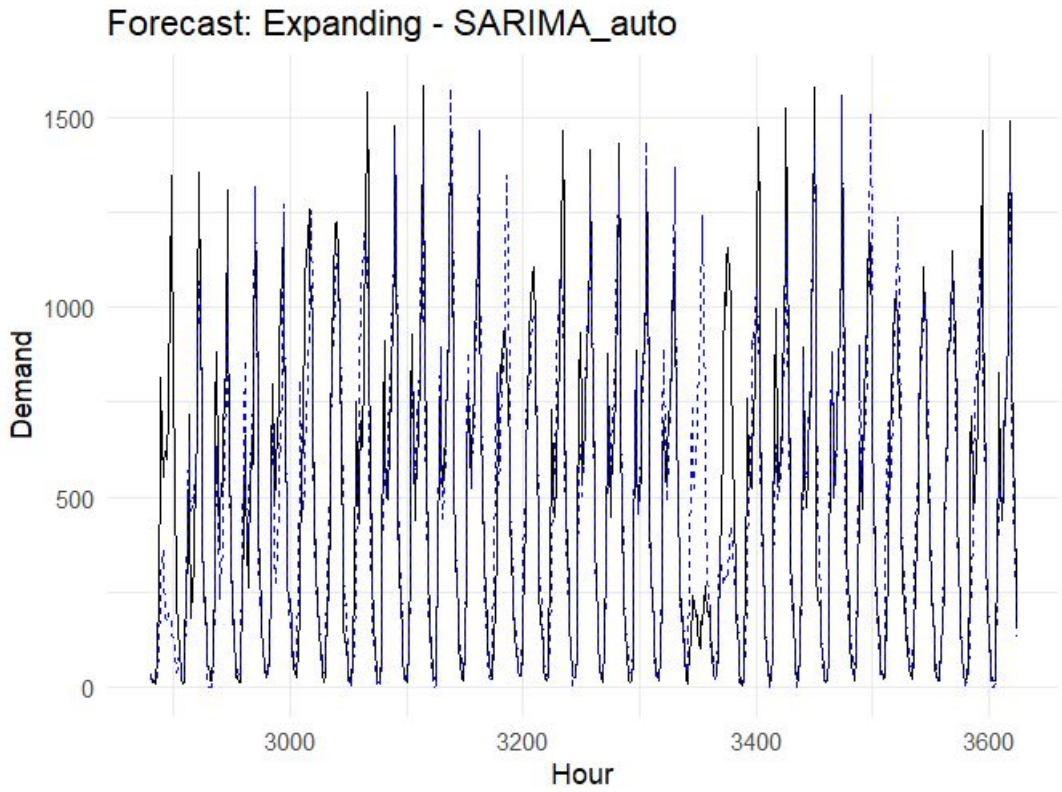

Forecast: Expanding - ARIMA_211
Demand
Hour
0
500
1000
1500
3000
3200
3400
3600

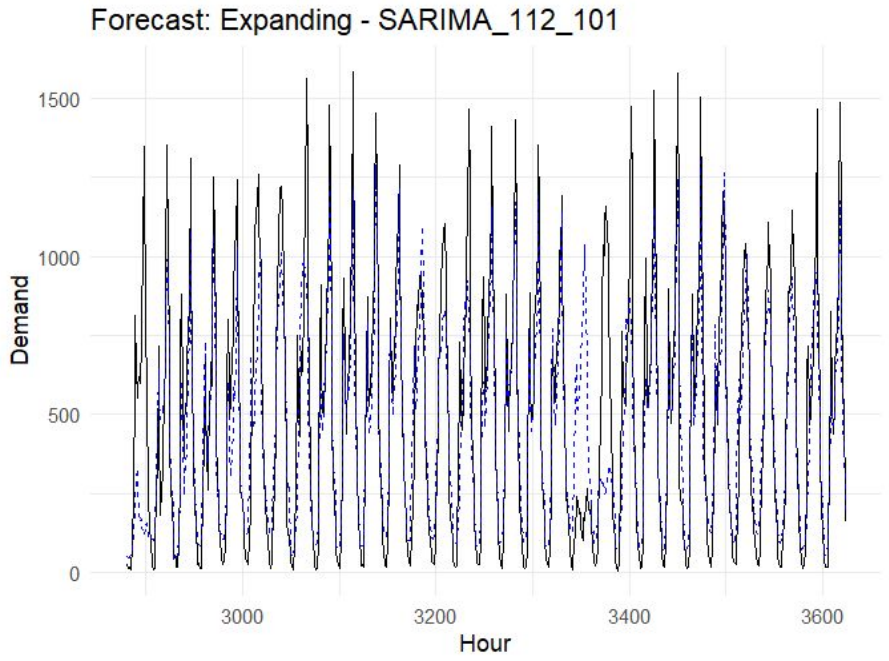

Forecast: Expanding - ARIMA_auto
Demand
Hour
0
500
1000
1500
3000
3200
3400
3600

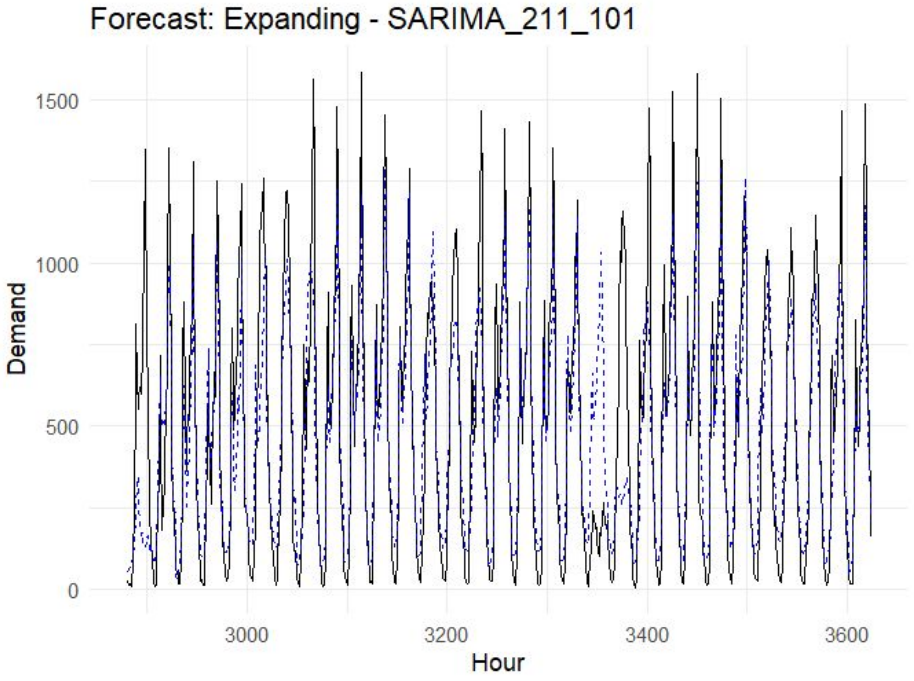

Forecast: Expanding - ARIMA_112
Demand
Hour
0
500
1000
1500
3000
3200
3400
3600

ARIMA(2,0,0)(2,1,0)

Added seasonality -> increase in forecast volatility



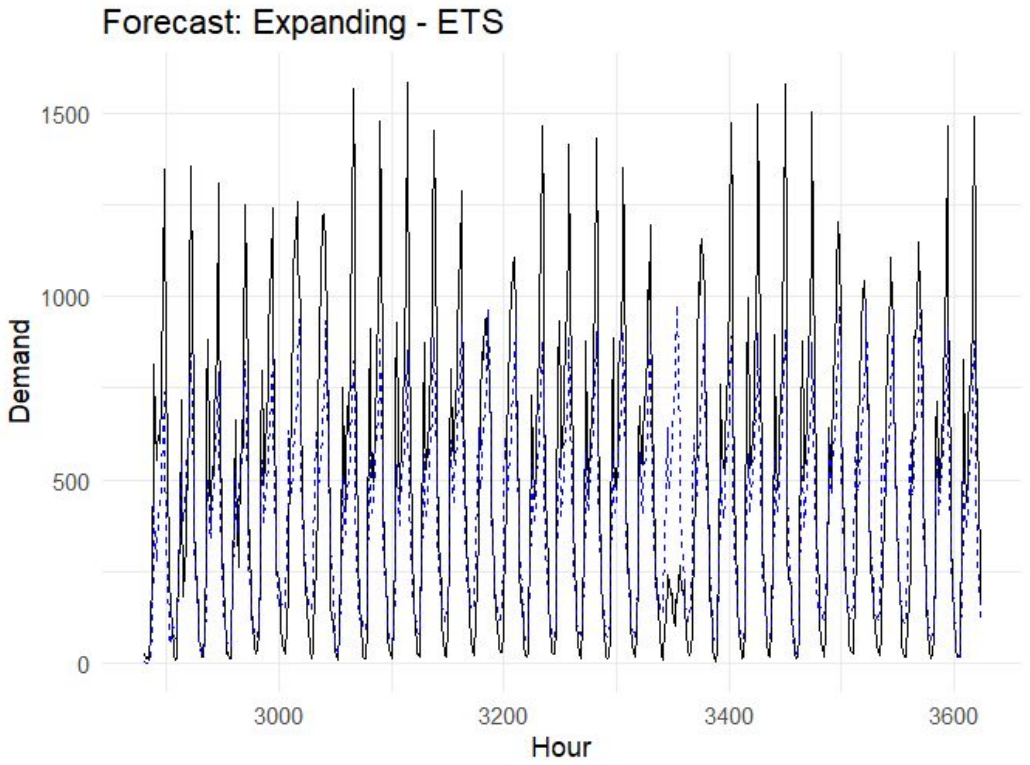

Forecast: Expanding - SARIMA_112_101
Demand
Hour
0
500
1000
1500
3000
3200
3400
3600


Forecast: Expanding - SARIMA_211_101
Demand
Hour
0
500
1000
1500
3000
3200
3400
3600

ETS(A,N,A)

Added seasonality -> increase in forecast volatility

Not as volatile as SARIMA, but better than ARIMA



Forecast comparisons

Some goodness-of-fit measures are obtained using accuracy().
For each model, accuracy() gives:

- ME – mean error (bias; closer to 0 is better);
- **RMSE** – root mean squared error (penalizes large errors; lower is better);
- **MAE** – mean absolute error (lower is better);
- MPE / MAPE – percentage errors;
- **MASE** – scaled error relative to a naïve benchmark (≈ 1: as good as naive; <1: better than naive; lower is better);
- ACF1 – lag-1 autocorrelation of residuals (closer to 0 is better; indicates a well-specified model.)

MASE

- Suitable for all three scenarios
- independent of scale of the data

$$q_t = \frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|}$$

Accuracy Measures

```
> print(do.call(rbind, rolling_accuracy))
                   MAE      RMSE      MASE       MAPE
ARIMA_111       406.2143 533.6058 4.541659 1.4366111
ARIMA_211       391.5938 512.9967 4.378196 1.6391233
ARIMA_112       405.3578 532.4059 4.532084 1.4464215
ARIMA_auto      393.3662 515.6371 4.398012 1.6097382
SARIMA_112_101        NA       NA       NA         NA
SARIMA_211_101 133.1034 202.6932 1.488156 0.9242559
SARIMA_auto     113.4205 194.4096 1.268092 0.3894219
ETS             160.6687 218.5463 1.796349 0.7687272
>
> cat("===== Expanding Accuracy =====\n")
===== Expanding Accuracy =====
> print(do.call(rbind, expanding_accuracy))
                   MAE      RMSE      MASE       MAPE
ARIMA_111       405.9664 533.2658 4.538888 1.4395765
ARIMA_211       391.6045 513.0320 4.378315 1.6384389
ARIMA_112       405.1745 532.1725 4.530034 1.4481532
ARIMA_auto      393.1576 515.3167 4.395679 1.6131223
SARIMA_112_101 132.9812 203.9313 1.486790 0.8472762
SARIMA_211_101        NA       NA       NA         NA
SARIMA_auto     113.4129 194.0199 1.268008 0.3863690
ETS             165.6197 223.8994 1.851703 0.7893692
>
```

Assessing significance of results

- Diebold-Mariano Test
- Model Confidence Set (MCS)

Diebold-Mariano test

A statistical technique which compares the accuracy of two forecasts. evaluates whether the predictive performance of two forecasting models differ significantly

- We test the Null hypothesis that the forecast errors coming from the two forecasts bring about the same loss.

Diebold-Mariano test procedure

- Compute the forecast errors:
- Choose loss function: In our case (**MASE)**
- Compute loss differential: $d_t \equiv [g(e_{it}) - g(e_{jt})]$
- Compute mean loss differential: $\bar{d} = \frac{1}{T}\sum_{t=1}^{T}[g(e_{it}) - g(e_{jt})]$
- Correct for serial correlation
- Compute DM statistic:

$$S_1 = \frac{\bar{d}}{\sqrt{\frac{2\pi \hat{f}_d(0)}{T}}},$$

Diebold Mariano test

```
===== DM Test (MASE Loss) =====
> print(dm_results)
                ARIMA_111     ARIMA_211     ARIMA_112    ARIMA_auto
ARIMA_111              NA  1.269116e-42  3.182718e-10  1.364309e-43
ARIMA_211    1.269116e-42            NA  1.163912e-41  6.834876e-32
ARIMA_112    3.182718e-10  1.163912e-41            NA  1.270907e-42
ARIMA_auto   1.364309e-43  6.834876e-32  1.270907e-42            NA
SARIMA_112_101 6.223482e-92  1.129747e-91  7.115650e-92  1.289027e-91
SARIMA_211_101 3.076105e-92  5.591428e-92  3.360112e-92  6.477447e-92
SARIMA_auto    4.244199e-94  5.147582e-93  5.155161e-94  4.485127e-93
ETS           4.861662e-105 5.335531e-105 5.790594e-105 7.122372e-105
              SARIMA_112_101 SARIMA_211_101  SARIMA_auto           ETS
ARIMA_111       6.223482e-92   3.076105e-92 4.244199e-94 4.861662e-105
ARIMA_211       1.129747e-91   5.591428e-92 5.147582e-93 5.335531e-105
ARIMA_112       7.115650e-92   3.360112e-92 5.155161e-94 5.790594e-105
ARIMA_auto      1.289027e-91   6.477447e-92 4.485127e-93 7.122372e-105
SARIMA_112_101            NA   2.917084e-01 5.359292e-14  9.614590e-10
SARIMA_211_101  2.917084e-01             NA 5.087367e-14  1.547015e-09
SARIMA_auto     5.359292e-14   5.087367e-14           NA  2.010399e-18
ETS             9.614590e-10   1.547015e-09 2.010399e-18            NA
>
```

Model Confidence Set

- A Statistical method of testing multiple forecasting models
- It identifies the best models by the their predictive performance

Model Confidence Set procedure

- Start with all models
- Compute loss for all models
- Compare using test statistic (T-MAX: maximum standardized loss difference)
- Bootstrap distribution
- Eliminate worse model
- Repeat until done
- We get MCS that cannot be rejected

Model Confidence set

```
===== MCS Results =====
> print(mcs_results)

------------------------------------------
-          Superior Set of Models        -
------------------------------------------
            Rank_M      v_M MCS_M Rank_R      v_R MCS_R     Loss
SARIMA_auto      1 -3.54047     1      1 -3.54047     1 1.257269

Details
------------------------------------------

Number of eliminated models     :       7
Statistic       :       Tmax
Elapsed Time    :       Time difference of 17.45311 secs
>
```

Final Recommendations

- Use both ADF and KPSS test
- Seasonality matters (also consider weekly seasonality for interpretation)
- Non-seasonal ARIMA not adequate
- Differencing affects interpretation