Report Presentation

EBS 2003: Second-year Project II

Obbe Pulles & Tobias Schnabel

July 1st, 2022

Table of Contents

Econometric Analysis

Exploratory Analysis

Cluster Identification

- Origin cluster: Cluster 1
- 4 most frequent destination clusters between 200 and 400km: clusters 12, 15, 186, 208
- Lanes aggregated to (week)daily data with n=204 obs. at periodicity 5

Outliers

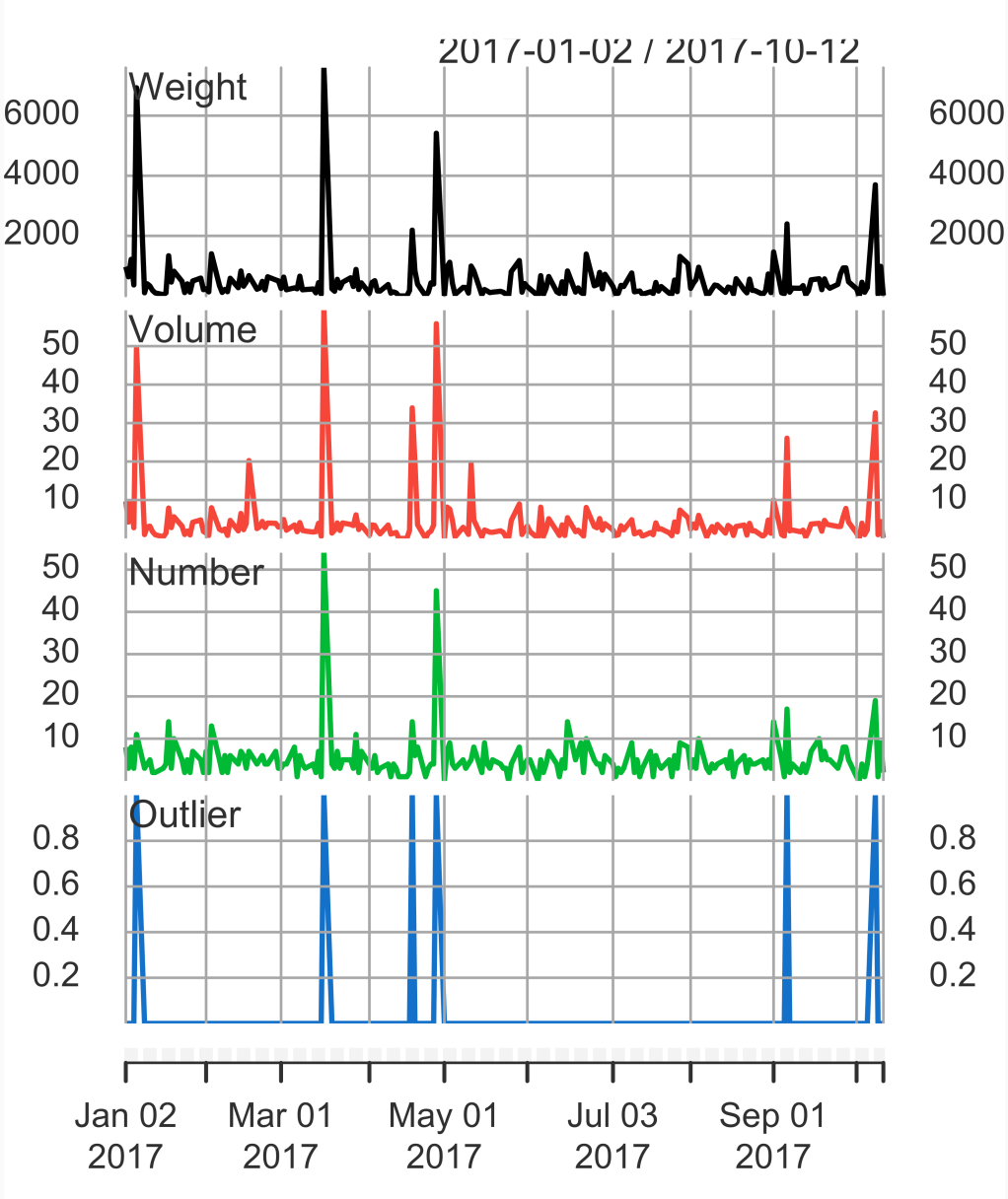
We use a custom function to generate dummy variable columns that are coded 1 for any observation where any of W_t, V_t, or N_t exceed their mean by 2 or more standard deviations, as given for *Weight* by

$$Flag_{Weight,i} = \begin{cases} 1 & \text{if } W_{t,i} \geq \mu_W + 2 \times \sigma_W \\ 0 & else \end{cases}$$

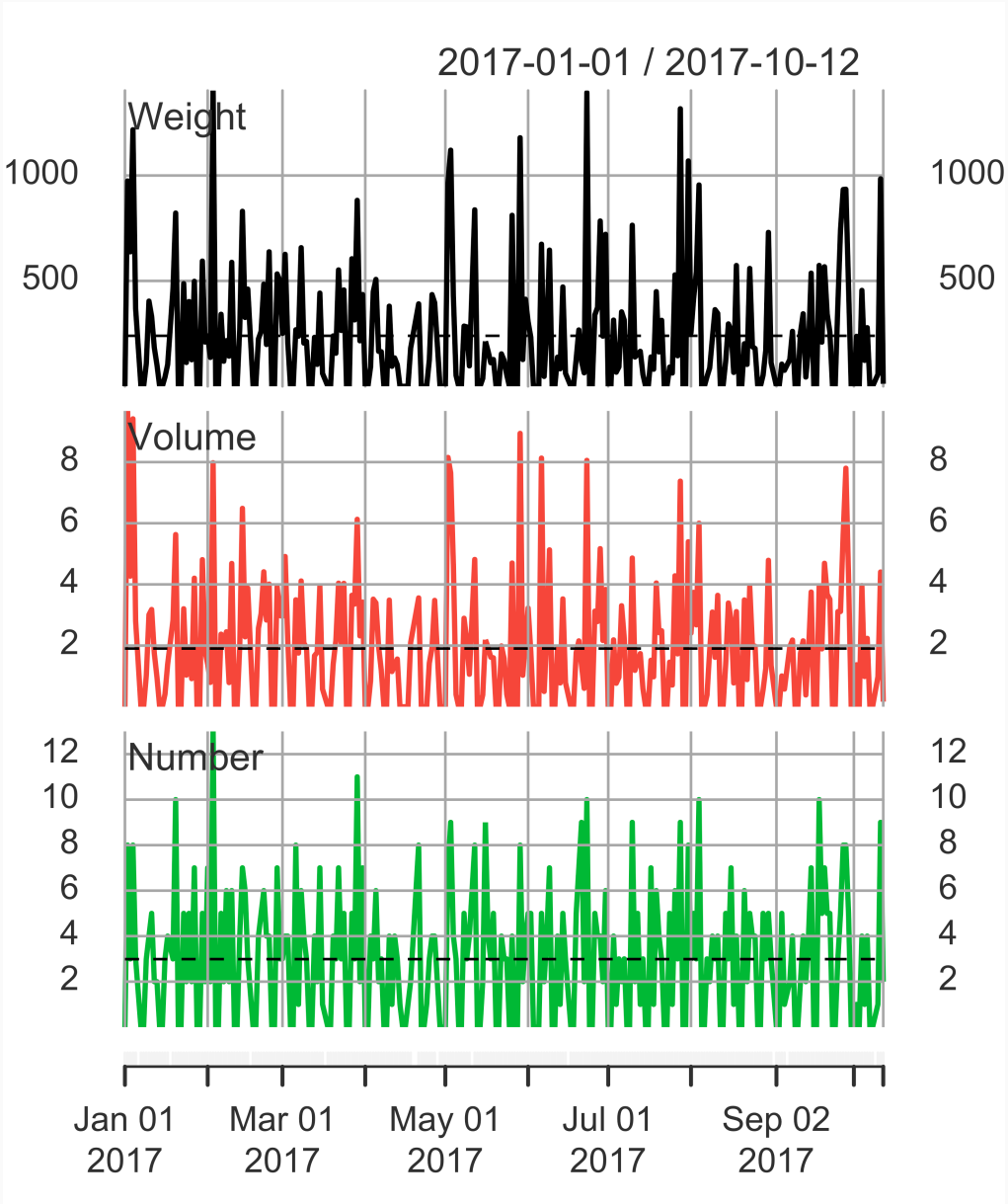
and use these three columns to create a fourth column:

$$Outlier = \begin{cases} 1 & \text{if } Flag_{Weight,i} = 1 \vee Flag_{Volume,i} = 1 \vee Flag_{Number,i} = 1 \\ 0 & \text{if } Flag_{Weight,i} = 0 \wedge Flag_{Volume,i} = 0 \wedge Flag_{Number,i} = 0 \end{cases}$$

Outliers (2)


2017-01-02 / 2017-10-12
Weight
6000
4000
2000
Volume
50
40
30
20
10
Number
Outlier
0.8
0.6
0.4
0.2
Jan 02
2017
Mar 01
2017
May 01
2017
Jul 03
2017
Sep 01
2017

Stationarity



Baseline Models

$$W_t = \alpha_0 + \alpha_1 Tuesday + \alpha_2 Wednesday + \alpha_3 Thursday + \alpha_4 Friday + \alpha_5 Outliers + \epsilon_t \quad (1)$$

$$V_t = \alpha_0 + \alpha_1 Tuesday + \alpha_2 Wednesday + \alpha_3 Thursday + \alpha_4 Friday + \alpha_5 Outliers + \epsilon_t \quad (2)$$

$$N_t = \alpha_0 + \alpha_1 Tuesday + \alpha_2 Wednesday + \alpha_3 Thursday + \alpha_4 Friday + \alpha_5 Outliers + \epsilon_t \quad (3)$$

Baseline Models: Estimation Results

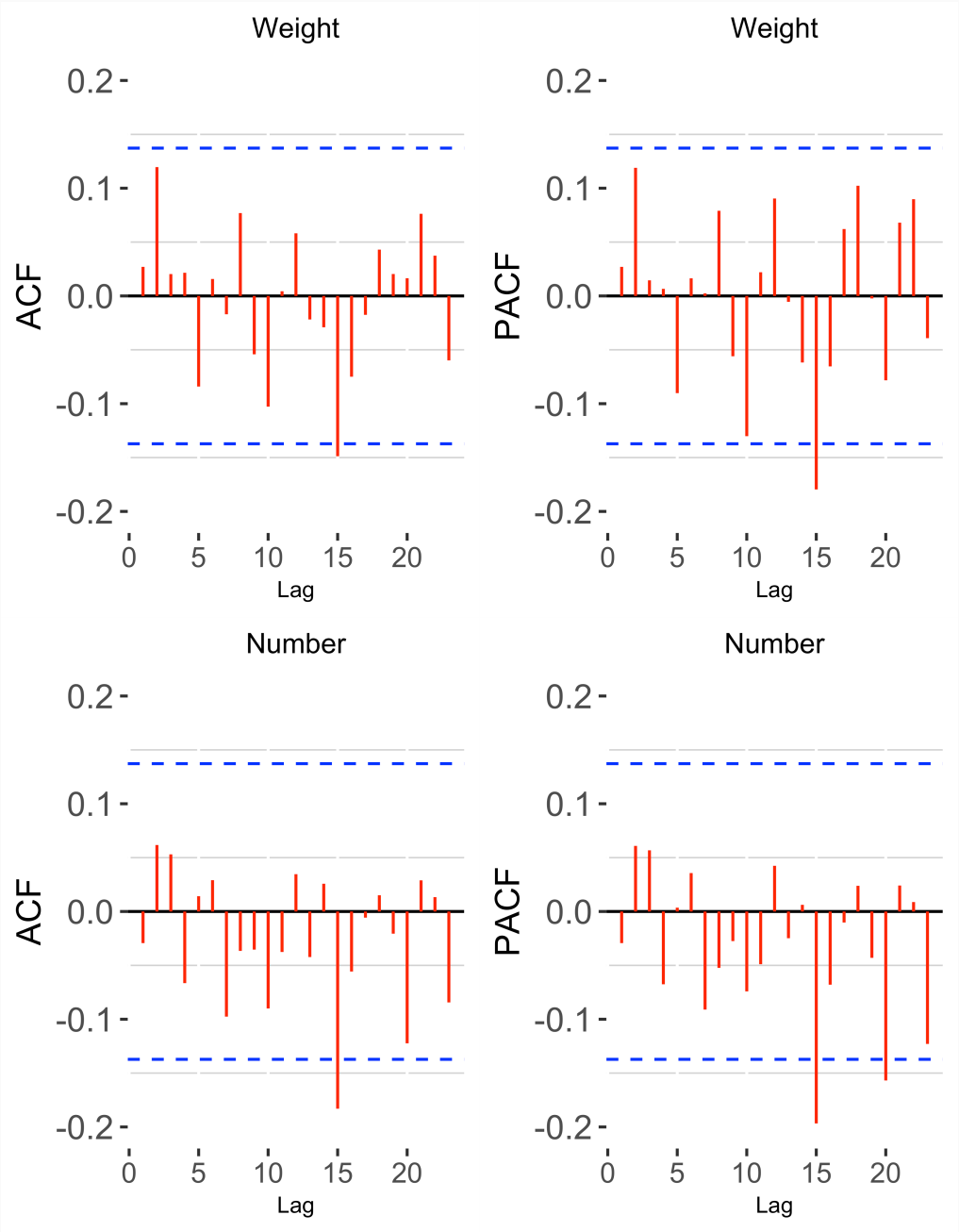
Table 1: Baseline Regression Results

	Dependent variable:		
	Weight	Volume	Number
tuesday	−21.0671	0.0723	−1.3210
wednesday	67.3354	0.5663	0.8576
thursday	−6.6085	0.8012	−1.3454*
friday	391.4009***	2.7827***	2.3422***
Outlier	4,204.3500***	39.0742***	20.8399***
Constant	280.7378***	2.2310***	4.3698***
Observations	204	204	204
Adjusted R^2	0.7276	0.7910	0.5411
F Statistic (df = 5; 198)	109.4428***	154.6670***	48.8782***
Note:			*p<0.1; **p<0.05; ***p<0.01

Econometric Analysis

SARMA(p,q)(P,Q) Models

SARMA(p,q)(P,Q)[5] Models: Identification



SARMA(p,q)(P,Q)[5] Models: Baseline Results

Table 2: ARMA(0,0)(0,0) Estimation Results

	Dependent variable:		
	Weight	Volume	Number
intercept	280.7378***	2.2310***	4.3698***
Outlier	4,204.3500***	39.0742***	20.8399***
tuesday	−21.0671	0.0723	−1.3210*
wednesday	67.3354	0.5663	0.8576
thursday	−6.6085	0.8012	−1.3454*
friday	391.4009***	2.7827***	2.3422***
Note:		*p<0.1; **p<0.05; ***p<0.01	

SARMA(p,q)(P,Q)[5] Models: Final Specifications

Models automatically selected based on AIC (BIC gives identical results)

$$\textit{Weight} : \textit{SARMA}(0,0,0)(2,0,1)[5]$$

$$\textit{Volume} : \textit{SARMA}(0,0,0)(0,0,0)[5]$$

$$\textit{Number} : \textit{SARMA}(0,0,0)(2,0,1)[5]$$

Table 3: Information Criteria Comparison

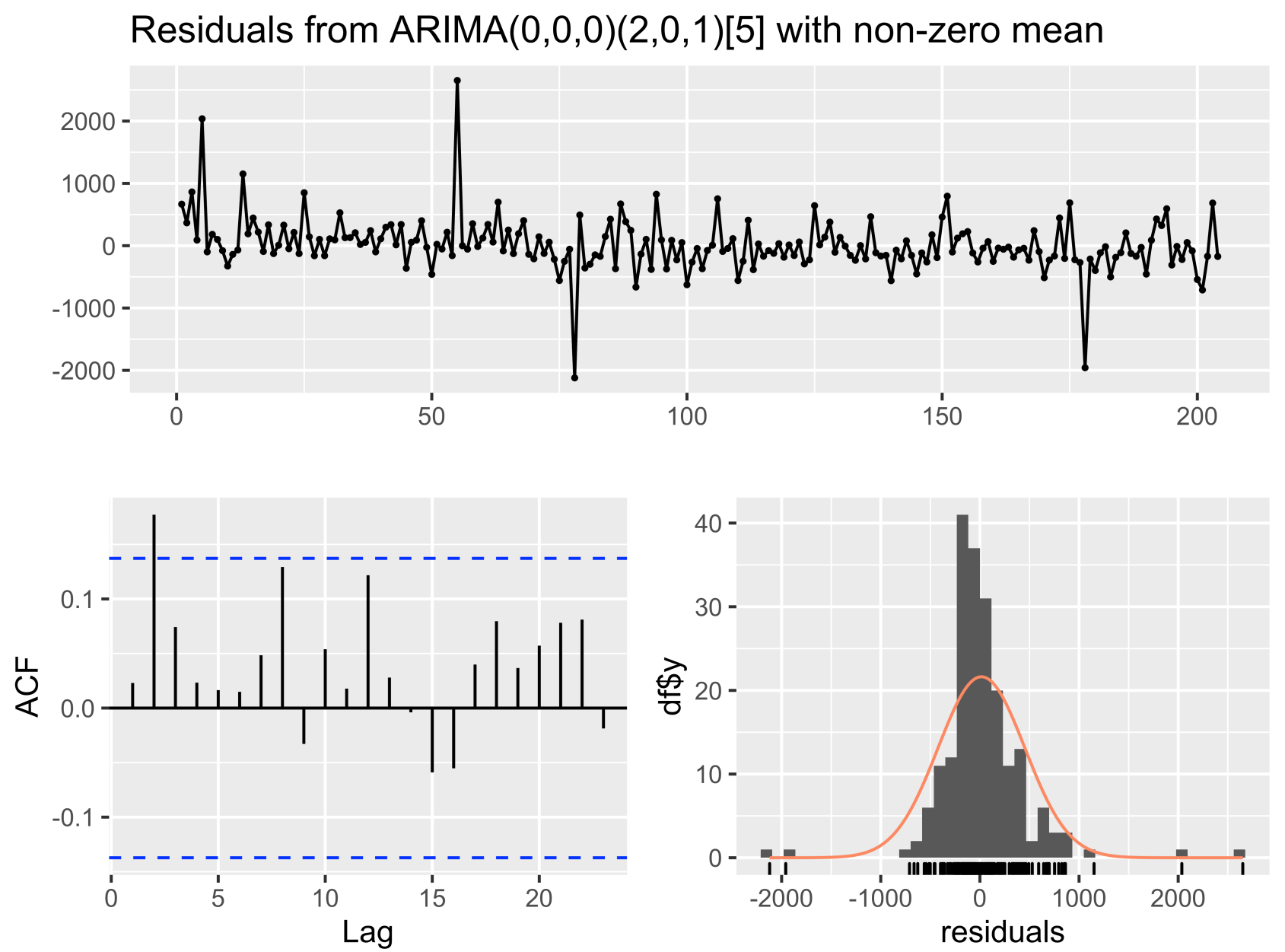
	BIC Baseline	BIC Final	AIC Baseline	AIC Final
Weight	15.242	15.281	15.128	15.086
Volume	5.493	5.493	5.379	5.379
Number	5.582	5.619	5.468	5.456

SARMA(p,q)(P,Q)[5] Models: Estimation Results

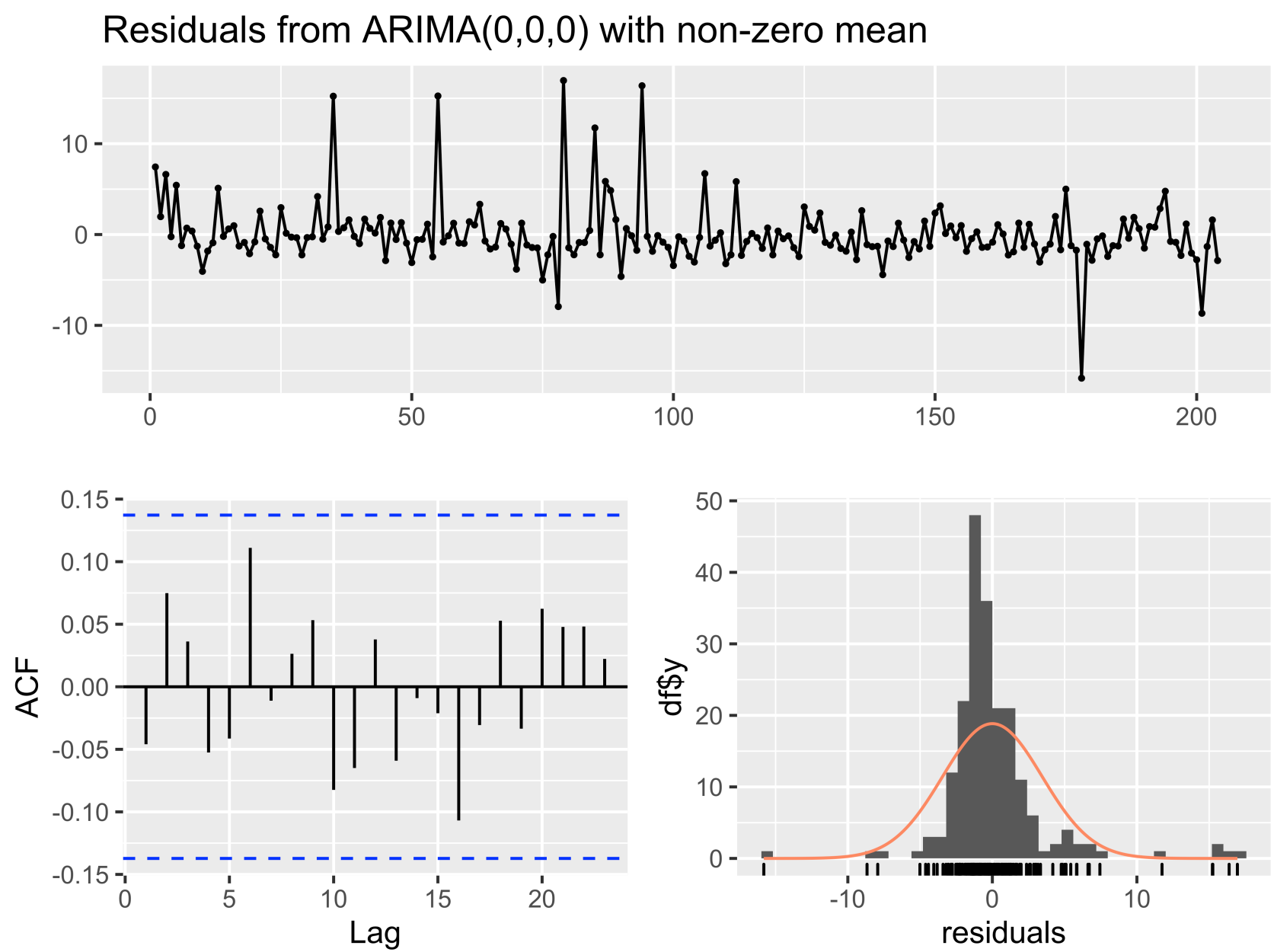
Table 4: ARMA(p,q)(P,Q) Estimation Results

	Dependent variable:		
	Weight	Volume	Number
sar1	0.5582***		0.6363***
sar2	−0.0988		−0.1783**
sma1	−0.7263***		−0.6549***
intercept	282.8321***	2.2310***	4.4237***
Outlier	4,151.9630***	39.0742***	21.0669***
tuesday	−25.5452	0.0723	−1.3545***
wednesday	39.9651	0.5663	0.6840
thursday	−8.1966	0.8012	−1.3919***
friday	380.5566***	2.7827***	2.4926***
Note:			*p<0.1; **p<0.05; ***p<0.01

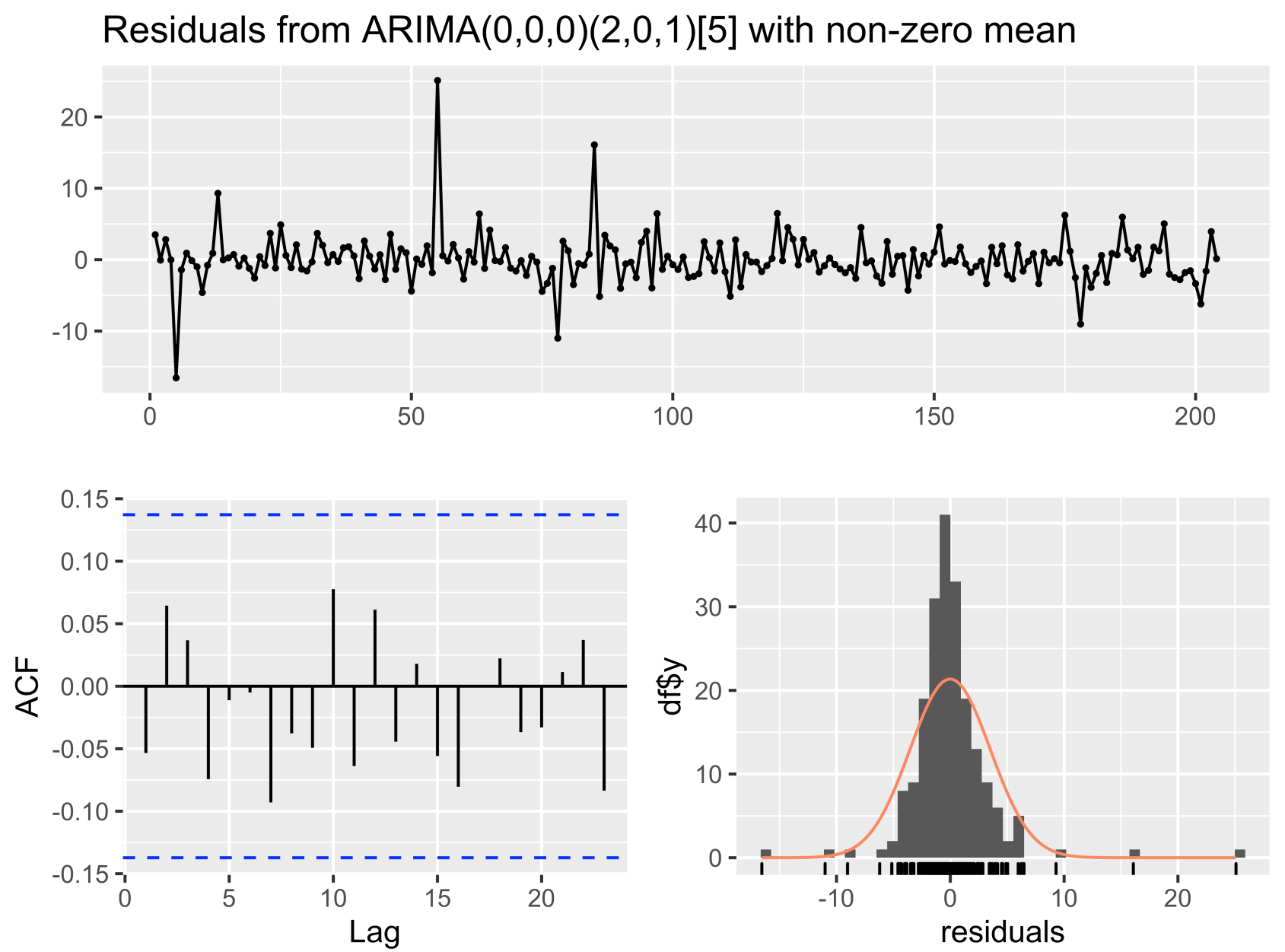
SARMA(p,q)(P,Q)[5] Models: Robustness



SARMA(p,q)(P,Q)[5] Models: Robustness (2)



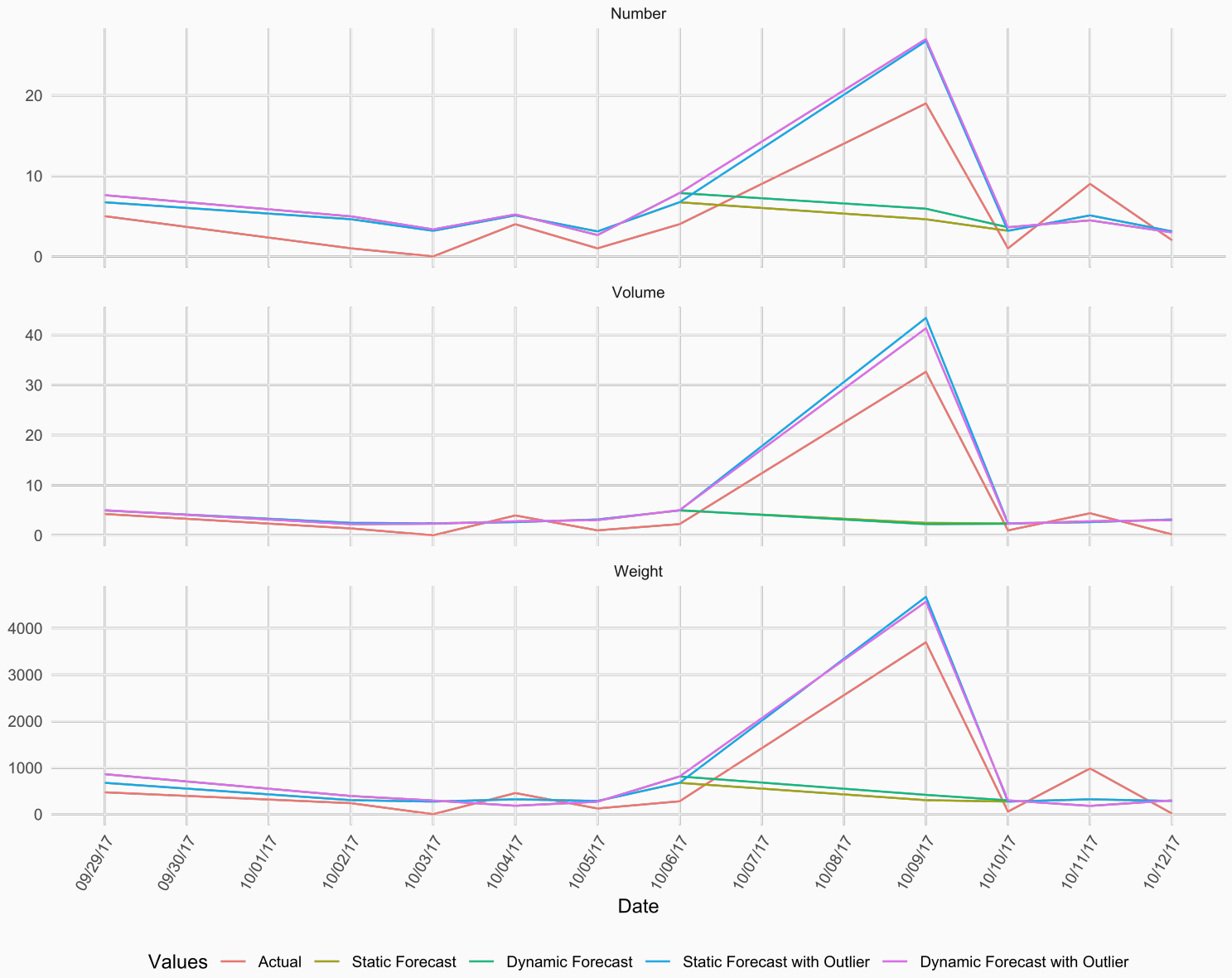
SARMA(p,q)(P,Q)[5] Models: Robustness (3)



Econometric Analysis

In-Sample Forecasting

Forecasting: In-Sample Accuracy



Forecasting: In-Sample Accuracy (2)

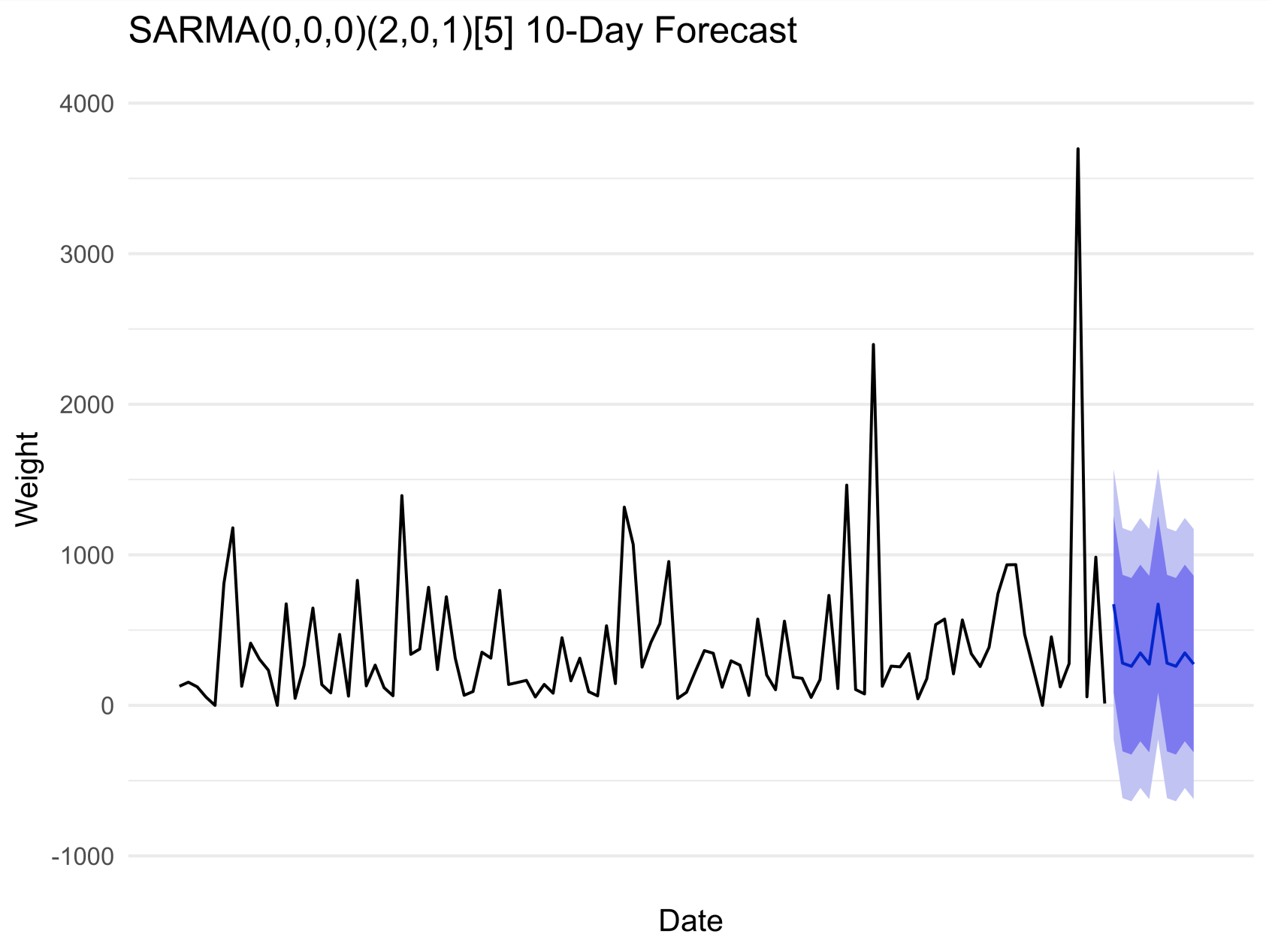
Table 5: RMSE of Forecast Models

	Weight	Volume	Number
Static Forecast	1113.878	9.721	5.171
Dynamic Forecast	1104.308	9.781	5.021
Static with Outlier	428.418	3.886	3.469
Dynamic with Outlier	468.031	3.281	3.810

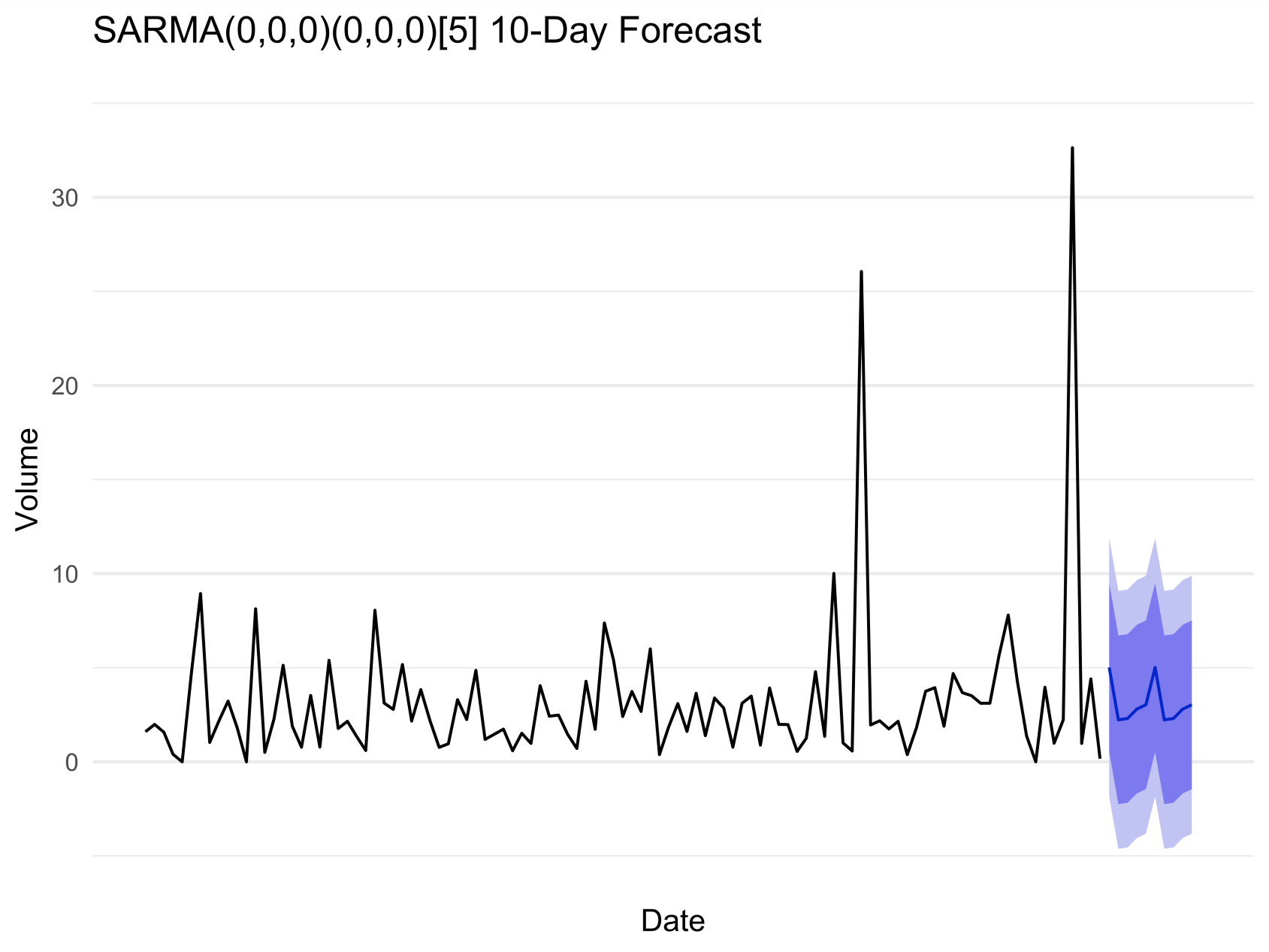
Econometric Analysis

Out-Of-Sample Forecasting

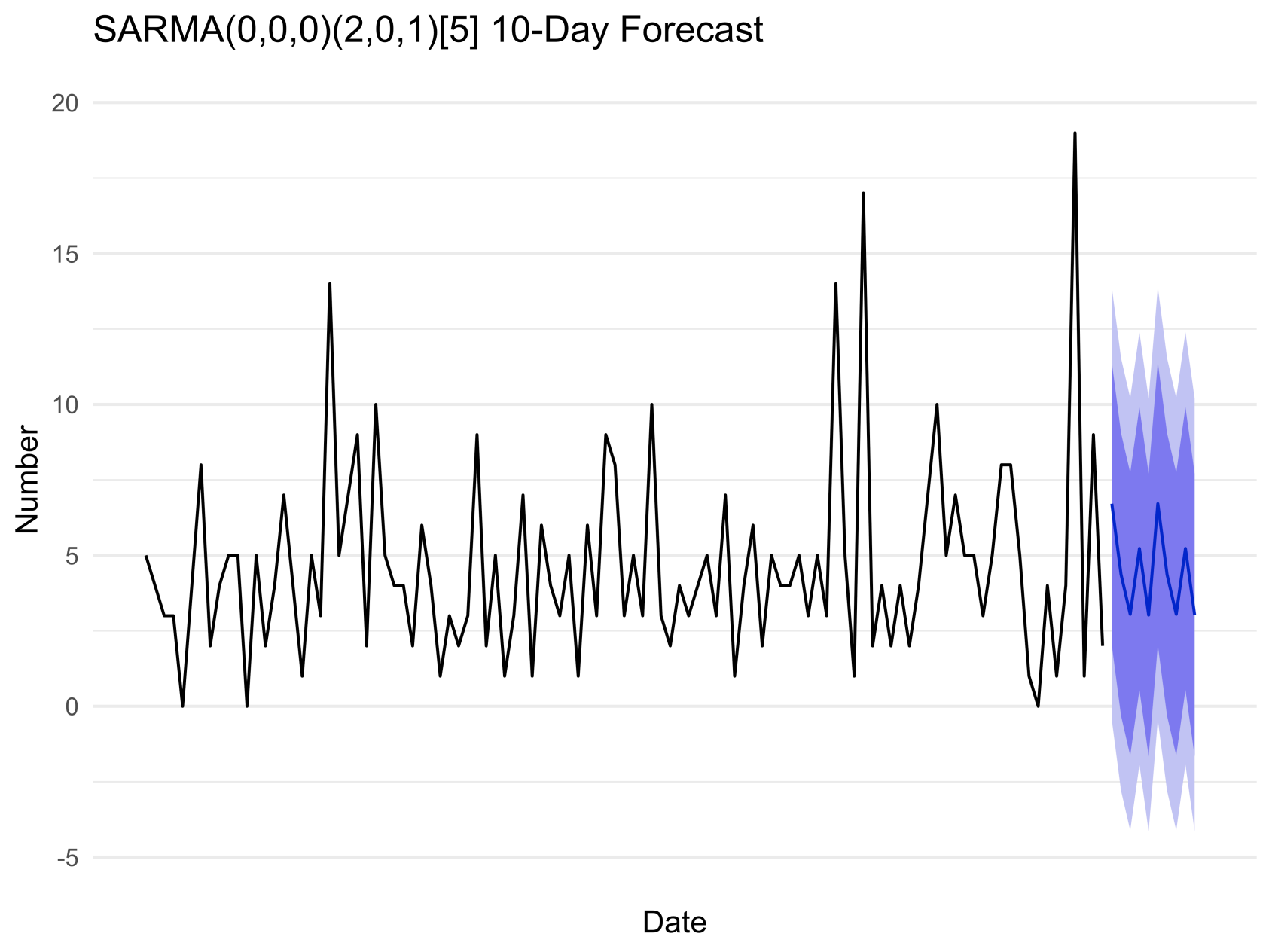
Forecasting: Out-of-Sample



Forecasting: Out-of-Sample (2)



Forecasting: Out-of-Sample (3)



Vehicle Routing

Program Structure

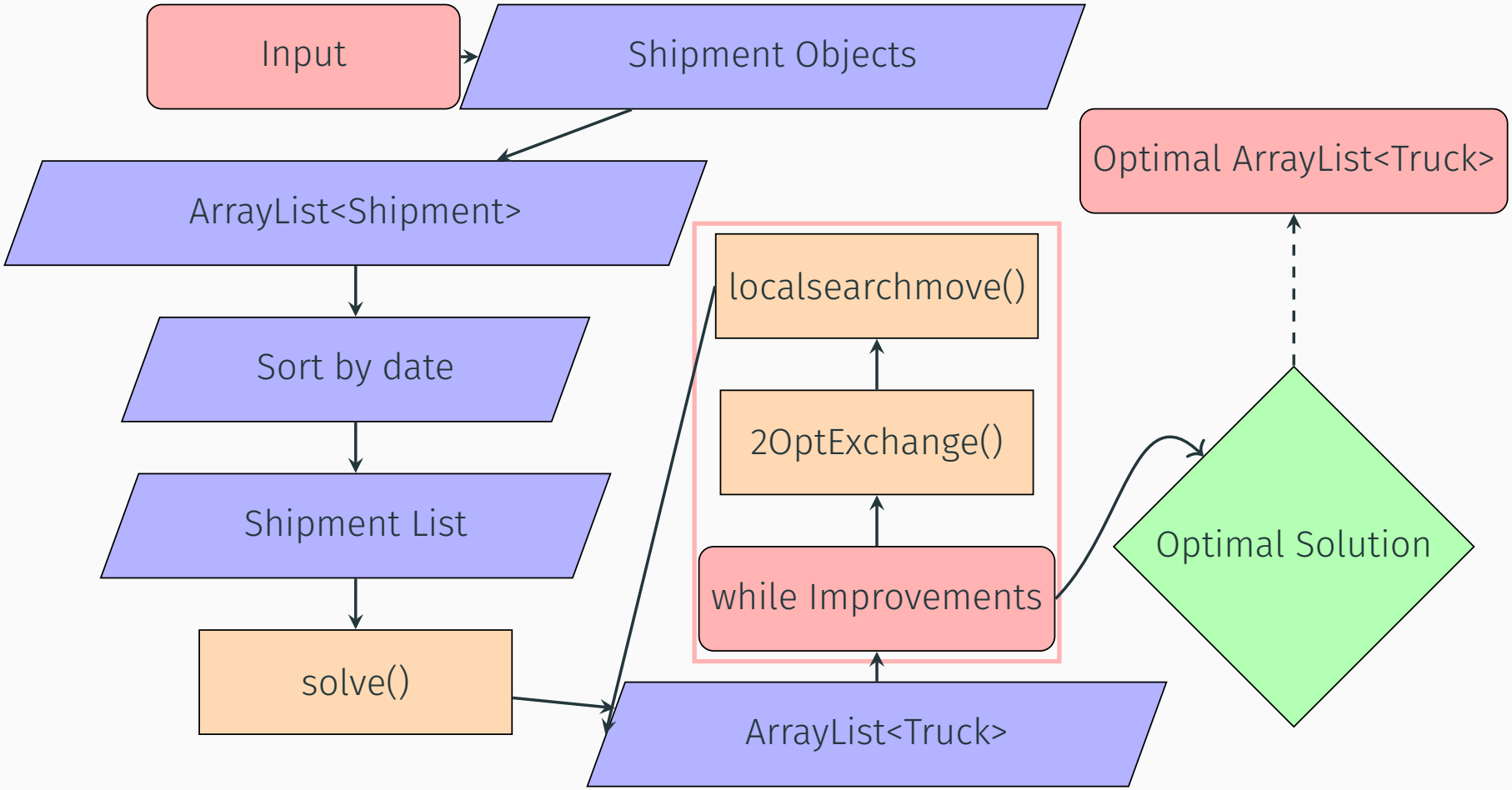
Objects

```
1  public class Shipment {
2      private final Date PDate;
3      private final double Weight;
4      private final double Volume;
5      private final double OriginClusterLat;
6      private final double OriginClusterLong;
7      private final Customer customer;
8      }
9
10 public class Customer {
11     private double lat;
12     private double lon;
13     private final String ID;
14     private int numShip = 0;
15     }
```

Objects (2)

```
1 public class Truck {
2     private double currentWeight = 0;
3     private double currentVolume = 0;
4     private final int truckId;
5     private final ArrayList<Shipment> shipments = new
      ArrayList<>();
6     private ArrayList<Customer> route = new ArrayList<>();
7     }
```

Execution Flowchart



Vehicle Routing

Local Search

Code Listing: Nearest Neighbour Heuristic

```
1  Let S ≠ ∅ be the set of shipments, C ≠ ∅ the set of customers;
2  Let T = {t_1} be the set of trucks;
3  while S ≠ ∅ do
4      Take t ∈ T;
5      Let c = minDistCustomer(t);
6      for s ∈ S do
7          Take s ∈ S;
8          if s.customer = c AND no constraints violated then
9              add(s, t), S = S\{s};
10             add(c, Route(t));
11         else if constraints violated then
12             T = T ∪ {t_new};
13             break;
```

Code Listing: 2-Opt-Exchange

```
1  Take t ∈ T a truck in the set of trucks;
2  Let R = Route(t), bd = distance(R);
3  Let newd = 0;
4  while foundImprovement do
5      for 1 ≤ i < |R| − 2 do
6          for i + 1 ≤ j < |R| − 1 do
7              newR = reverse(R, i, j);
8              newd = distance(newR);
9              if newd < bd then
10                 bd = newd;
11                 R = newR;
```

Code Listing: Local Search Move

```
1  Take t_1, t_2 ∈ T, where t_1 ≠ t_2 are trucks;
2  Let R_1 =Route(t_1), R_2 =Route(t_2), bd = distance(R_1 + R_2);
3  while foundImprovement do
4      for 1 ≤ i < |R_1| − 1 do
5          for 1 ≤ j < |R_2| − 1 do
6              Let c = R_2[j] gets customer in route 2 at position j;
7              Let newR_1 = R_1 ∪ {c} at position i;
8              Let newR_2 = R_2\{c};
9              if distance(newR_1 + newR_2) <bd AND no constraints
                are violated then
10                 bd = totalDistiance(newR_1 + newR_2);
11                 Move all shipments with customer c from t_2 to t_1;
12                 R_1 = newR_1, R_2 = newR_2;
```

Vehicle Routing

Results

Results

```
Date #10: 24/07/2017
Total cost of the feasible basic solution: 1493.39 EUR
Truck #1: Weight: 15.8/22 Volume: 78.77/82
Route: Cluster > FR2055 > FR1207 > FR2006 > FR321 > FR34 >
     FR178 > FR4710 > FR5000 > FR3307 > FR743 > FR47 >
    FR813 > FR3601 > FR2834 > FR1890 > FR953 > FR >
    Cluster

Truck #2: Weight: 3.6/22 Volume: 62.24/82
Route: Cluster > FR > Cluster

Truck #3: Weight: 2.5/22 Volume: 38.34/82
Route: Cluster > FR2754 > FR2845 > FR144 > FR3236 > FR2672
     > FR3191 > FR849 > FR226 > Cluster

Optimal Total cost : 1491.58 EUR
Number of trucks used: 3
Method executed in 6 milliseconds, of which
Obtaining the feasible basic solution took 2 ms
Optimizing the basic solution took 4 ms
```

Questions?