PATH Forecast Documentation

Planning & Regional Development

6/3/2020

DRAFT

Summary

This note attempts to document the process used to produce long-range PATH forecasts. Different sections are designed for different audiences in the hope of providing measures of transparency and reproducability.

Discussion across internal agency staff regarding the PATH forecasts generally employ the term "model." There are actually three models, one apiece for weekday, Saturday and Sunday ridership, used to develop raw projections of future ridership.[1]

Each model is trained using data on economic conditions, seasonal conditions, ridership disruptions (events), and historic ridership. The forward-looking forecasting process then proceeds by subjecting future scenarios for those economic and seasonal conditions to those three equations; this produces predicted values for future ridership.

The product of this process is an Excel file containing predicted ridership, all data used as inputs in the process, and diagnostics commonly reported as part of statistical modeling.

This file's second section embeds most of the relevant source code included in the process outlined above, and the full code will be provided as an attached source file when material is transmitted to PATH in the future.

PATH and Planning developed the general outlines for the models three years ago. Specifications since then have not changed, outside of the allowance of shifting time series controls.[2] The weekday model, which easily forecasts the largest share of total ridership, has performed generally well but the weekend models have gradually lost predictive power as weekend closures have provided near-useless data points in increasing frequency. PATH and Planning attempted to address this challenge in 2019 and it remains unresolved.

The only other major change of note since the 2017 model development process is a shift in the definition of day types. Holidays were at one point all classified and treated as Sundays; the shift to the current treatment had a minor impact on the forecast.

The file has four sections. First, the summary (above). Second, code documentation. This is included for transparency and reproducability, and can be moved to a later section in subsequent versions; for now, it is easiest to keep it as the second section. It can be used, for example, to find what file names are imported to and exported from the process. Data on ridership is read directly from a file provided by PATH. Data on days per month is also provided by PATH and is pasted into a larger file containing days and variables for various events or cyclical modeling treatments, including but not limited to weekend closures, Hurricane Sandy, and seasonal (monthly) variation. Third, output. This focuses on annual forecasted ridership. Fourth, modeling statistics and diagnostics. Modelers can review this information at their interest.

[1] Holidays are distributed across weekdays (for minor holidays, where PATH has found similarities between weekday and minor holiday ridership behavior) and Saturdays (major holidays). Weekdays include one minor holiday apiece in October (Columbus Day) and November (Veterans Day). Saturdays include other holidays. Sundays represent only Sunday ridership.

[2] The models' time series controls are reset roughly once a year, with guidance provided by automated variable selection algorithms. Nerds, see: https://otexts.com/fpp2/arima-r.html

Code

The economics data, already converted to monthly, is combined with days, dummy variables (simulating past events that need special statistical controls), ridership, and fares. They're saved across a handful of small worksheets and are merged in-memory. The data is thus almost completely unaltered prior to actual analysis (model estimation and forecasting). The exception is the fare variable, which is converted from nominal to real using the most recent macroeconomic forecast's value for national CPI.

PATH forecasts through early 2020 used regional CPI, and going forward will use national CPI to conform with TB&T forecasting and agency financial analysis.

A key manual option in the model is to select the months where (1) ridership data ends and, immediately following that, (2) the forecasting process begins:

```
end = "2020-02-01"
end_and_one = "2020-03-01"
```

The modeling process employs ridership data directly from a file provided by PATH and attaches it to other files that include information on days, fare, economic variables and other data points.

```
path = read.csv("./input data/PATH_PaxCounts_2000-2009+2010-2020Apr.csv")
path$month = as.Date(paste(path$year, str_pad(path$month, 2,
    pad = "0"), "01", sep = "-"))
path$year = NULL
```

Table 1: Sample: PATH ridership file pt 1

month	avg_wkdayholminor_tstile	avg_satholmajor_tstile	avg_sun_tstile
2000-01-01	232710	85325.67	60462
2000-02-01	244019	94431.80	65227
2000-03-01	247544	99765.00	68909
2000-04-01	245914	100145.00	67410
2000-05-01	257985	92609.40	75643
2000-06-01	260820	102604.00	77873

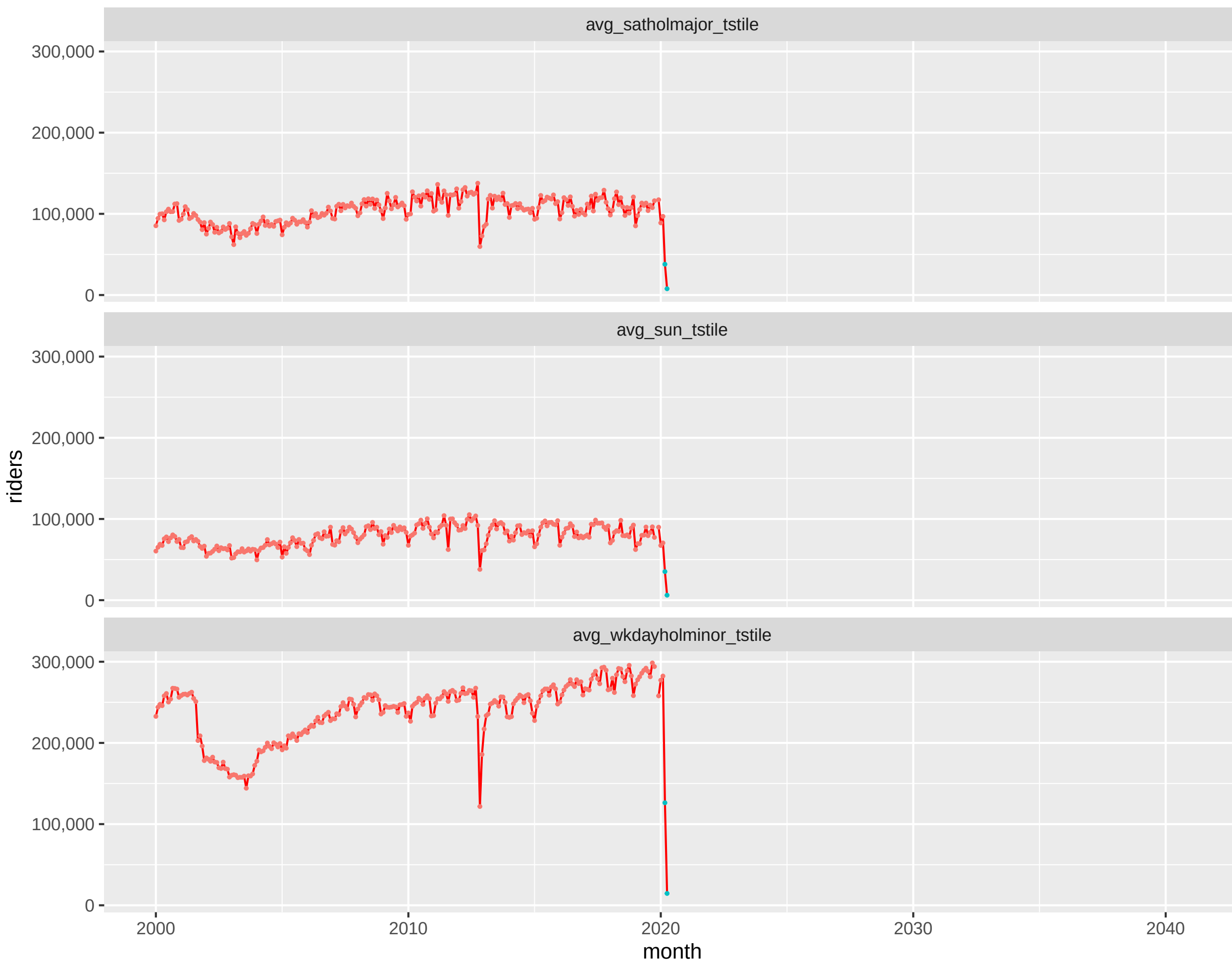
Table 2: Sample: PATH ridership file pt 2

num_wkdayholminor	num_satholmajor	num_sun	total_days
20	6	5	31
20	5	4	29
23	4	4	31
20	5	5	30
22	5	4	31
22	4	4	30

Details for calculation of real fare in footnote.[3]

```
path$cpi_base = path[path$month == end, "cpi_2020_06"]
path$real_farefare = ifelse(path$month <= end, path$fare_nominal *
    path$cpi_base/path$cpi_2020_06, max(path$fare_nominal))
path$cpi_base = NULL
```

PATH ridership has grown at roughly 1.2% annually for the past three decades, including dips (following 9/11 and Hurricane Sandy) and jumps (significant weekday growth in the years preceding 2019):

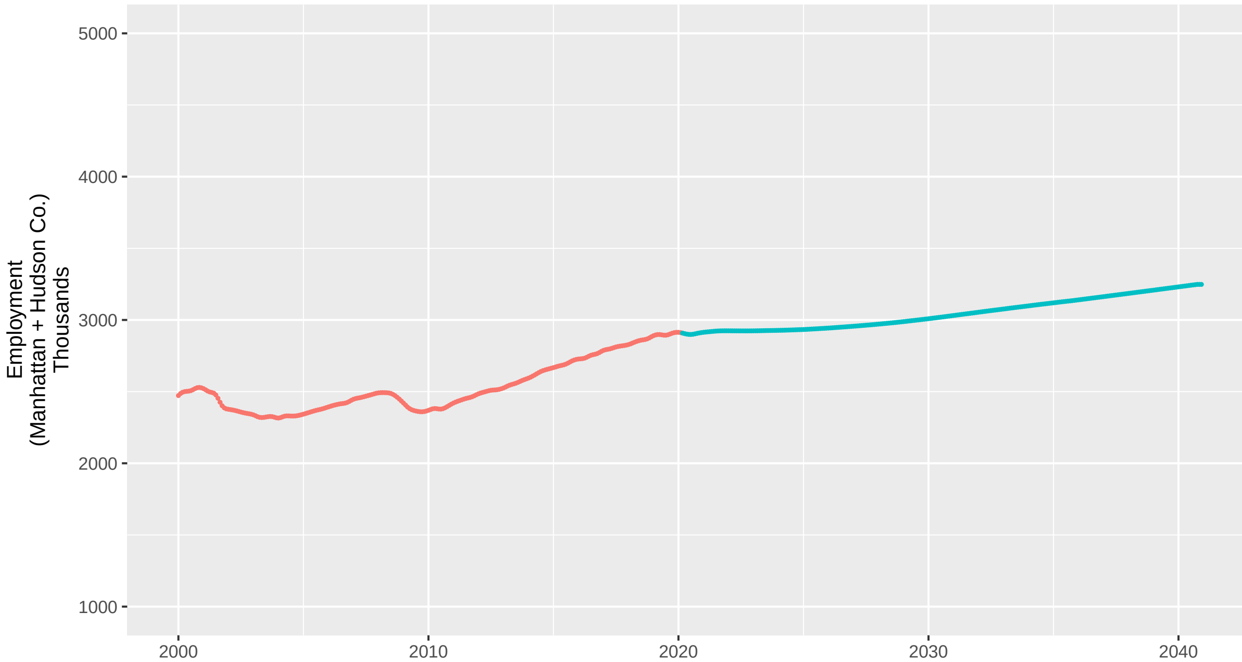


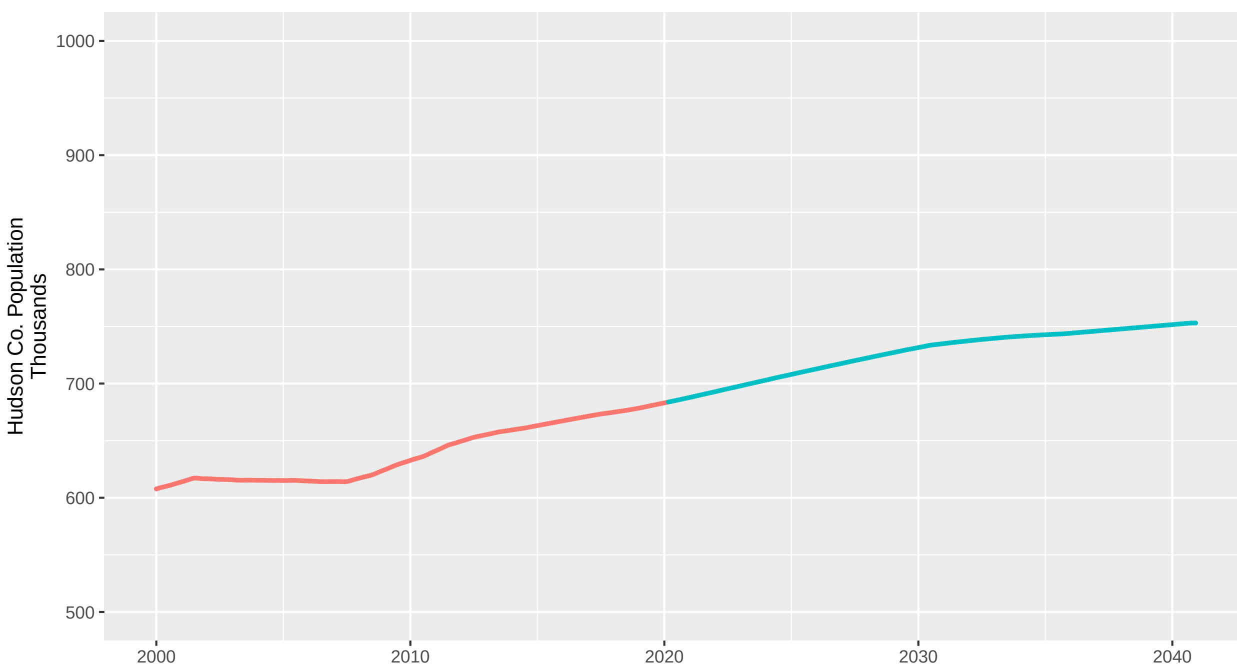

```
before = subset(path, path$month <= end & path$month >= "2004-01-01")  #before = head(path, 218)
after = subset(path, path$month > end)  #after = tail(path, 250)
```

The code for training the models are below, starting with a quick list of the variables chosen for the three equations, one apiece for weekday, Saturday and Sunday ridership. Note that the Saturday and Sunday equations use the same varibles. Key predictors are Manhattan and Hudson County employment for the weekday model and Hudson County population for the weekend models.

[3]https://stackoverflow.com/questions/25646333/code-chunk-font-size-in-rmarkdown-with-knitr-and-latex/57151528#57151528

The model specifications followed a joint 2017 project that included a broad review of potential predictors. Employment at the county level (Manhattan and Hudson County, combined) is not only a strong predictor of weekday ridership, it also more stable than other variables with roughly equivalent predictive power.





Red Points Represent Data, Blue Represent Forecast

Models are trained below

Weekday:

```
fit = arima(ts(before$avg_wkdayholminor_tstile), xreg = oldreg,
    order = c(0, 0, 1), include.mean = T)
```

Saturday:

```
fitsat = arima(ts(before$avg_satholmajor_tstile), xreg = oldregsat,
    order = c(1, 1, 0))
```

Sunday:

```
fitsun = arima(ts(before$avg_sun_tstile), xreg = oldregsat, order = c(1,
    1, 1))
```

Model equation coefficients and residuals are found at the end of this file. Subsequent code forecasts future ridership and organizes results for export:

```
pathpredict = predict(fit, n.ahead = forec_horizon, newxreg = newreg)  # level=95 #interval = 'predicti
pathpredictsat = predict(fitsat, n.ahead = forec_horizon, newxreg = newregsat)
pathpredictsun = predict(fitsun, n.ahead = forec_horizon, newxreg = newregsat)
pathpredict_pess = predict(fit_pess, n.ahead = forec_horizon,
    newxreg = newreg_pess)
pathpredict_opt = predict(fit_opt, n.ahead = forec_horizon, newxreg = newreg_opt)
```

```
pathpredict_by_month = as.data.frame(cbind(pathpredict$pred,
    pathpredictsat$pred, pathpredictsun$pred, pathpredict_pess$pred,
    pathpredict_opt$pred))
names(pathpredict_by_month) = c("avg_wkdayholminor_tstile", "avg_satholmajor_tstile",
    "avg_sun_tstile", "pess_wkdayholminor", "opt_wkdayholminor")
pathpredict_by_month$month = seq(as.Date(end) + extra, as.Date(future),
    by = "mon")
```

```
# Annual
pathpredict_by_month$year = year(pathpredict_by_month$month)
pathpredict_year = summaryBy(sum_wkdayholminor + sum_satholmajor +
    sum_sun + sum_wkday_pess + sum_wkday_opt ~ year, data = pathpredict_by_month,
    FUN = sum)
names(pathpredict_year) = c("year", "base_wkday", "saturday",
    "sunday", "pess_wkday", "opt_wkday")
pathpredict_year$base_total = pathpredict_year$base_wkday + pathpredict_year$saturday +
    pathpredict_year$sunday
pathpredict_year$pess_total = pathpredict_year$pess_wkday + pathpredict_year$saturday +
    pathpredict_year$sunday
pathpredict_year$opt_total = pathpredict_year$opt_wkday + pathpredict_year$saturday +
    pathpredict_year$sunday
pathpredict_by_month$year = NULL
```

```r
pathpredict_year = pathpredict_year[c(1, 2, 3, 4, 7, 5, 8, 6,
    9)]
```

```r
resids = as.data.frame(cbind(as.vector(resid(fit)), as.vector(resid(fitsat)),
    as.vector(resid(fitsun))))
names(resids) = c("Weekday_residuals", "Saturday_residuals",
    "Sunday_residuals")
```

Output

Table 3: Annual Results

year	base_wkday	saturday	sunday	base_total
2,020	73,230,694	7,057,312	4,709,045	84,997,051
2,021	73,351,361	7,487,885	4,879,864	85,719,110
2,022	73,740,148	7,553,363	4,973,182	86,266,694
2,023	73,516,938	7,711,241	5,160,690	86,388,869
2,024	74,232,797	7,872,852	5,173,671	87,279,319
2,025	74,215,143	8,023,413	5,255,195	87,493,751
2,026	74,603,373	8,163,457	5,337,524	88,104,355
2,027	74,767,715	8,456,438	5,419,340	88,643,493
2,028	75,590,304	8,490,264	5,610,204	89,690,772
2,029	76,222,975	8,638,200	5,605,516	90,466,691
2,030	76,940,788	8,769,805	5,681,422	91,392,015
2,031	77,715,330	8,853,356	5,724,965	92,293,650
2,032	78,484,161	9,062,595	5,762,546	93,309,302
2,033	79,229,916	9,004,258	5,795,416	94,029,590
2,034	79,653,286	9,050,125	5,933,620	94,637,032
2,035	80,659,354	9,091,030	5,861,570	95,611,954
2,036	81,712,454	9,135,266	5,884,496	96,732,216
2,037	82,160,971	9,180,582	5,913,072	97,254,625
2,038	82,925,458	9,240,879	5,947,204	98,113,540
2,039	83,695,399	9,301,327	5,981,421	98,978,147
2,040	84,454,579	9,360,600	6,015,009	99,830,188

Table 4: Scenarios

year	pess_wkday	opt_wkday	saturday	sunday	pess_total	opt_total
2,020	72,639,675	74,322,709	7,057,312	4,709,045	84,406,032	86,089,066
2,021	71,396,789	75,020,718	7,487,885	4,879,864	83,764,538	87,388,467
2,022	71,499,785	76,059,323	7,553,363	4,973,182	84,026,330	88,585,869
2,023	71,262,949	76,111,738	7,711,241	5,160,690	84,134,880	88,983,669
2,024	71,850,176	76,999,847	7,872,852	5,173,671	84,896,698	90,046,369
2,025	71,921,981	77,112,756	8,023,413	5,255,195	85,200,589	90,391,364
2,026	72,401,495	77,591,794	8,163,457	5,337,524	85,902,477	91,092,776
2,027	72,625,167	77,804,276	8,456,438	5,419,340	86,500,945	91,680,053
2,028	73,422,298	78,633,830	8,490,264	5,610,204	87,522,766	92,734,298
2,029	73,962,526	79,182,623	8,638,200	5,605,516	88,206,242	93,426,339
2,030	74,539,551	79,771,201	8,769,805	5,681,422	88,990,779	94,222,428
2,031	75,145,945	80,384,339	8,853,356	5,724,965	89,724,265	94,962,660
2,032	75,789,096	81,039,189	9,062,595	5,762,546	90,614,237	95,864,330
2,033	76,430,246	81,702,366	9,004,258	5,795,416	91,229,920	96,502,040
2,034	76,760,804	82,035,673	9,050,125	5,933,620	91,744,549	97,019,418
2,035	77,684,964	83,004,431	9,091,030	5,861,570	92,637,564	97,957,031
2,036	79,008,141	84,381,303	9,135,266	5,884,496	94,027,903	99,401,065
2,037	79,356,063	84,726,418	9,180,582	5,913,072	94,449,716	99,820,072
2,038	80,012,942	85,401,616	9,240,879	5,947,204	95,201,024	100,589,698
2,039	80,673,925	86,081,019	9,301,327	5,981,421	95,956,673	101,363,768
2,040	81,337,143	86,762,745	9,360,600	6,015,009	96,712,752	102,138,355

Save everything as:

```
sheets = list(Data = path, Monthly_Output = pathpredict_by_month,
    Annual_Output = pathpredict_year, Residuals = resids)
write_xlsx(sheets, "./output data/Output 2020-06.xlsx")  # This exports and names the file.
```

Monthly output included in Excel file within output folder.

Modeling statistics and diagnostics

Weekday

Table 5: Weekday Coefficients

term	estimate	std.error
ma1	0.5833455	0.0474794
intercept	-74147.5470399	24090.0379464
before.man__hud	134.0146557	14.1326237
before.dummy__2	5130.8414599	3359.5457699
before.dummy__3	7718.4628431	4481.7682504
before.dummy__4	13175.5625832	4505.4501544
before.dummy__5	16029.9753210	4502.2766282
before.dummy__6	19445.0540735	4500.3133018
before.dummy__7	15495.7335724	4499.3635453
before.dummy__8	9472.1385175	4499.2345499
before.dummy__9	19943.7981424	4499.7387264
before.dummy__10	17505.3643117	4500.9568985
before.dummy__11	10557.5296428	4587.5787981
before.dummy__12	334.7519945	3440.3617506
before.dum__911__base	-23489.4756153	11455.2807013
before.supersandy	-74428.8123894	10222.6727578
before.real__farefare	-13235.9292754	6513.4862965

Table 6: Weekday Diagnostics

sigma	logLik	AIC	BIC
11031.05	-2070.799	4177.598	4236.326

Table 7: Weekday Additional Diagnostics

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	47.81592	11031.05	8311.184	-0.2797387	3.623178	1.110952	0.2937603

Saturday

Table 8: Saturday Coefficients

term	estimate	std.error
ar1	-0.5585557	0.0605882
before.pop_hudson	525.5225232	656.4662690
before.dummy_2	6786.3400467	1793.2610958
before.dummy_3	20289.2641746	1657.9716433
before.dummy_4	22280.6951319	1994.4527931
before.dummy_5	14117.4337628	1988.5983538
before.dummy_6	21114.4686677	2107.4877997
before.dummy_7	17170.7508602	2082.0337264
before.dummy_8	15859.4409280	2119.9038154
before.dummy_9	16961.8121742	2000.1710572
before.dummy_10	20740.4709997	2016.7326045
before.dummy_11	15022.6709464	1702.5676902
before.dummy_12	15648.1692574	1858.0100782
before.dum_911_base	863.8195830	6277.7255044
before.supersandy	-44293.1117216	4170.8290772
before.end_close	-12901.1422792	2123.8095206
before.real_farefare	-24869.4627473	9719.1619472

Table 9: Saturday Diagnostics

sigma	logLik	AIC	BIC
6072.247	-1945.433	3926.866	3985.501

Table 10: Saturday Additional Diagnostics

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	60.12152	6056.512	4728.306	-0.1730156	4.460554	0.557294	-0.1645165

Sunday

Table 11: Sunday Coefficients

term	estimate	std.error
ar1	-0.0444697	0.0876576
ma1	-0.7803199	0.0507866
before.pop_hudson	348.9650880	188.9499864
before.dummy_2	5191.8040161	1561.8280465
before.dummy_3	8015.9956978	1572.1323548
before.dummy_4	14028.5018301	1584.5244531
before.dummy_5	18530.2740739	1590.2614219
before.dummy_6	23939.9748781	1594.3078541
before.dummy_7	17487.9940047	1596.0601539
before.dummy_8	15940.1578108	1599.1169934
before.dummy_9	20933.8705973	1595.1289709
before.dummy_10	17504.5206604	1588.7563584
before.dummy_11	14106.1979802	1618.8688041
before.dummy_12	16600.6895264	1610.2462863
before.dum_911_base	-2977.6993300	5031.4350444
before.supersandy	-37399.0166332	3680.7318596
before.end_close	-10379.7458646	1411.2464146
before.real_farefare	-12312.8866481	5569.5973032

Table 12: Sunday Diagnostics

sigma	logLik	AIC	BIC
4882.671	-1903.723	3845.445	3907.338

Table 13: Sunday Additional Diagnostics

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	204.6204	4870.015	3560.297	-0.0641905	4.549521	0.5180502	-0.0101541

Model residuals