BHao\_Project1

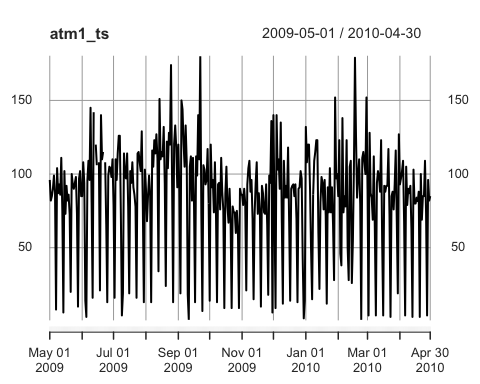
## Project 1: Part A

First, we load the data and convert to time series with a 7-day frequency. We'll start with ATM1.

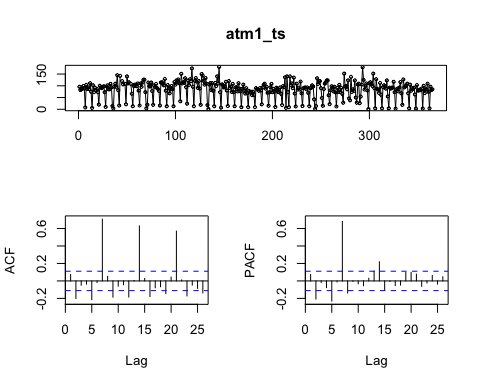
atm = read.csv('/Users/brucehao/Google Drive/CUNY/git/DATA624/Project 1/ATM624Data.csv', stringsAsFactors = F)  
atm$DATE = as.Date(atm$DATE, '%m/%d/%Y')  
atm1 = atm %>% filter(ATM=='ATM1') %>% select(DATE, Cash)  
atm1\_ts = as.xts(atm1$Cash, order.by = atm1$DATE)  
attr(atm1\_ts, 'frequency') = 7

Next, we plot the data to see what we're dealing with. There does not appear to be any particular trend, and there's insufficient data to tell if there is annual seasonality. But there is cearly a weekly pattern with significant dips on Thursdays. The 7-day pattern is clearly captured by the ACF and PACF plots below. Interestingly, the tsdisplay and forecast charts plot the week the x-axis rather than the date.

plot(atm1\_ts)



tsdisplay(atm1\_ts)



Interestingly, it appears that the weekly dip shifts from Thursdays to Tuesdays in mid February 2010. We also see similar shifts for the other ATMs. Assuming nothing corrupted the data like a faulty ETL process, we might assume that a weekly process that put the ATMs out of order for the much of the day (maybe maintenance, replishment, etc.) changed from Thursdays to Tuesdays in mid February 2010. Lastly, in the case of ATM2, it appears as if Wednesday cash volumes were also reduced after mid February.

atm\_all = atm %>% filter(ATM != '') %>%   
 mutate(Weekday = factor(weekdays(DATE), levels = c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')),   
 Period = if\_else(DATE < '2010-2-14', 'Pre', 'Post'))   
  
atm\_all %>%  
 group\_by(ATM, Period, Weekday) %>%   
 summarise(avg = mean(Cash, na.rm = T)) %>%   
 ggplot(aes(x = Weekday, y = avg, fill = Period)) +   
 geom\_bar(stat = 'identity', position = 'dodge') +   
 facet\_grid(ATM ~ ., scales = 'free')

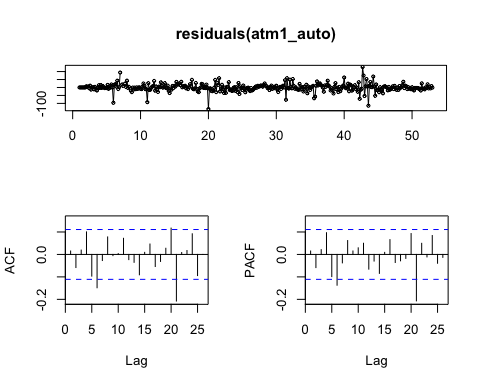


Auto.arima with D = 1 to prevent convergence suggests an ARIMA(1,0,0)(2,1,0) model that still produces 1-2 spikes in the ACF and PACF residuals plots, but 1 can be expected given the number of data points.

atm1\_auto = auto.arima(atm1\_ts, D = 1)  
summary(atm1\_auto)

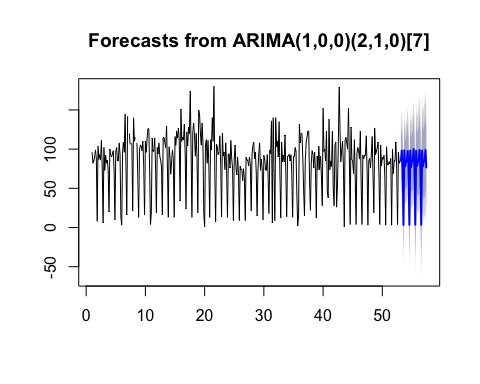
## Series: atm1\_ts   
## ARIMA(1,0,0)(2,1,0)[7]   
##   
## Coefficients:  
## ar1 sar1 sar2  
## 0.1511 -0.4698 -0.2048  
## s.e. 0.0535 0.0517 0.0514  
##   
## sigma^2 estimated as 601.5: log likelihood=-1641.05  
## AIC=3290.11 AICc=3290.22 BIC=3305.63  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -0.08937139 24.28804 15.37384 -98.72979 115.434 0.8697514  
## ACF1  
## Training set 0.009201676

tsdisplay(residuals(atm1\_auto))



Lastly, we forecast the month of May 2010 using the model that auto.arima suggested. The forecast captures the mean well, and it also nicely captures the dip and correctly places it in Tuesdays (rather than Thursdays) going forward.

atm1\_forecast = forecast(atm1\_auto, h = 31)  
plot(atm1\_forecast)

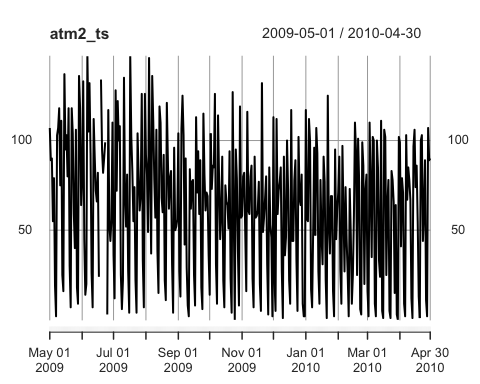


write.csv(atm1\_forecast$mean, '/Users/brucehao/Google Drive/CUNY/git/DATA624/Project 1/atm1\_forecast.csv')

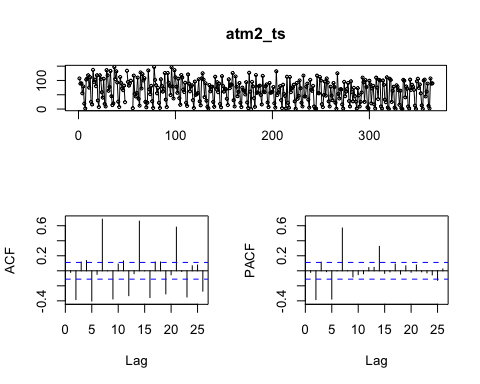
Okay, now we'll move on to ATMs 2 and 4. We'll handle ATM3 separately since there is so little data with which to build a model.

ATM2:

atm2 = atm %>% filter(ATM=='ATM2') %>% select(DATE, Cash)  
atm2\_ts = as.xts(atm2$Cash, order.by = atm2$DATE)  
attr(atm2\_ts, 'frequency') = 7  
plot(atm2\_ts)



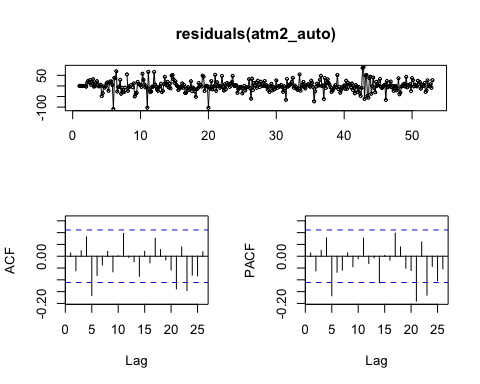
tsdisplay(atm2\_ts)



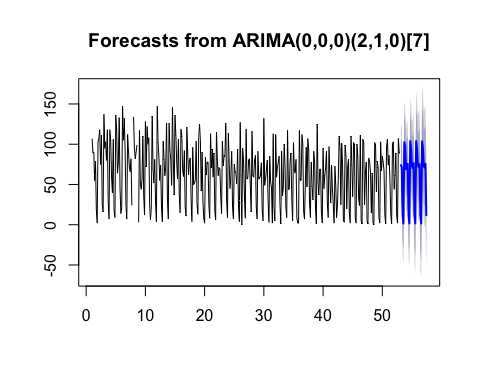
atm2\_auto = auto.arima(atm2\_ts, D = 1) #, stepwise = F, approximation = F)  
summary(atm2\_auto)

## Series: atm2\_ts   
## ARIMA(0,0,0)(2,1,0)[7]   
##   
## Coefficients:  
## sar1 sar2  
## -0.5747 -0.1766  
## s.e. 0.0521 0.0518  
##   
## sigma^2 estimated as 649.2: log likelihood=-1659.31  
## AIC=3324.61 AICc=3324.68 BIC=3336.26  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -0.1179362 25.23195 17.27829 -Inf Inf 0.8434248 0.01502367

tsdisplay(residuals(atm2\_auto))



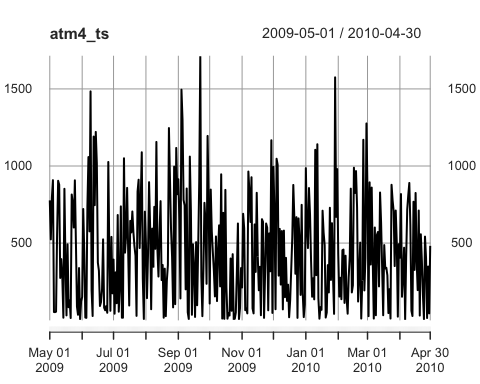
atm2\_forecast = forecast(atm2\_auto, h = 31)  
plot(atm2\_forecast)



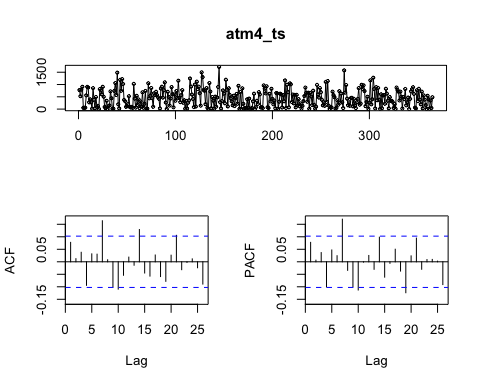
write.csv(atm2\_forecast$mean, '/Users/brucehao/Google Drive/CUNY/git/DATA624/Project 1/atm2\_forecast.csv')

ATM4: There is a clear outlier on 2/9/2010 which is an order of magnitude higher than any other day. Since it occurs right around the shift from Tuesday dips to Thursday dips that we discussed above, it may not be faulty data, but it definitely throws off our forecast. So we will use the average Tuesday cash volume prior to the shift.

atm4 = atm %>% filter(ATM=='ATM4') %>% select(DATE, Cash)   
atm4\_subCash = atm\_all %>% filter(Weekday == 'Tuesday', Period == 'Pre', Cash < 1000) %>%  
 summarise(avg = mean(Cash)) %>% as.numeric()  
atm4[atm4$DATE == '2010-02-09', 'Cash'] = atm4\_subCash  
atm4\_ts = as.xts(atm4$Cash, order.by = atm4$DATE)  
attr(atm4\_ts, 'frequency') = 7  
plot(atm4\_ts)



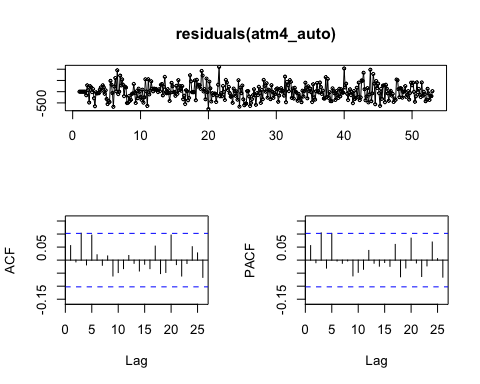
tsdisplay(atm4\_ts)



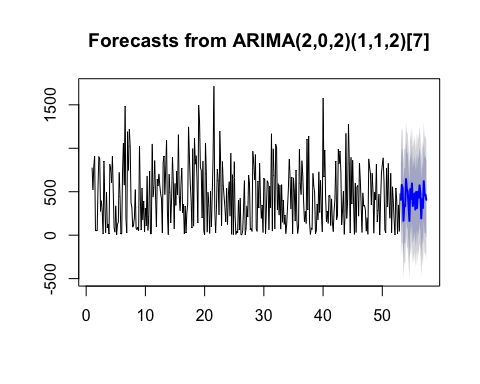
atm4\_auto = auto.arima(atm4\_ts, D = 1) #, stepwise = F, approximation = F)  
summary(atm4\_auto)

## Series: atm4\_ts   
## ARIMA(2,0,2)(1,1,2)[7]   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 sar1 sma1 sma2  
## 0.7119 -0.9375 -0.6770 0.976 -0.8796 -0.0577 -0.7804  
## s.e. 0.0307 0.0418 0.0245 0.036 0.2519 0.2687 0.2490  
##   
## sigma^2 estimated as 110746: log likelihood=-2590.24  
## AIC=5196.48 AICc=5196.89 BIC=5227.52  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -2.644705 326.3404 251.5571 -457.9961 495.5724 0.7260819  
## ACF1  
## Training set 0.05570104

tsdisplay(residuals(atm4\_auto))



atm4\_forecast = forecast(atm4\_auto, h = 31)  
plot(atm4\_forecast)



write.csv(atm4\_forecast$mean, '/Users/brucehao/Google Drive/CUNY/git/DATA624/Project 1/atm4\_forecast.csv')

ATM3: Since ATM3 has only 3 data points that exactly match ATM1's for the same dates, we'll just set ATM3's forecast to the same as ATM1's.

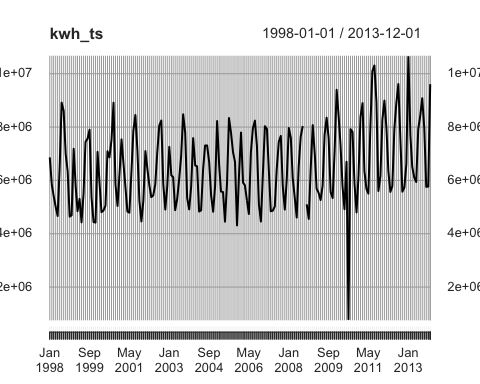
## Project 1: Part B

Again, we first load the data and configure it as a time series with a 12 month frequency.

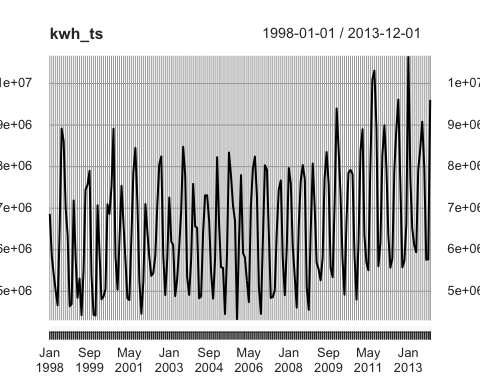
kwh = read.csv('/Users/brucehao/Google Drive/CUNY/git/DATA624/Project 1/ResidentialCustomerForecastLoad-624.csv', stringsAsFactors = F)  
kwh$Date = as.Date(paste0(kwh$YYYY.MMM, '-01'), '%Y-%b-%d')  
kwh\_ts = as.xts(kwh$KWH, order.by = kwh$Date)  
attr(kwh\_ts, 'frequency') = 12

Then, we take a preliminary look at the data. We see that data for September 2008 is NA, so we'll impute it by using the average for other Septembers (even though we recognize that there is a slight upward trend to the data and September 2008 is toward the latter half of the data set). We'll do the same thing for July 2010, which is a clear outlier.

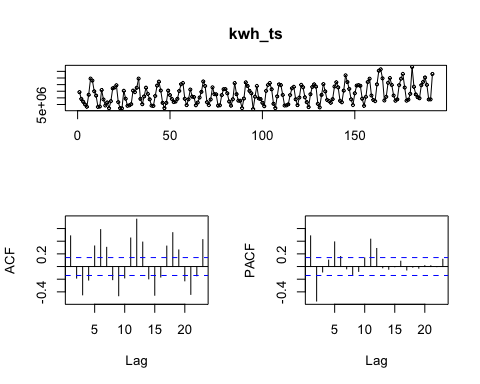
plot(kwh\_ts)



kwh\_sep08 = kwh %>% filter(month(Date) == 9) %>% summarise(avg = mean(KWH, na.rm = T)) %>% as.numeric()  
kwh\_ts['2008-09-01'] = kwh\_sep08  
kwh\_jul10 = kwh %>% filter(month(Date) == 7, year(Date) != 2010) %>%   
 summarise(avg = mean(KWH, na.rm = T)) %>% as.numeric()  
kwh\_ts['2010-07-01'] = kwh\_jul10  
  
plot(kwh\_ts)



tsdisplay(kwh\_ts)

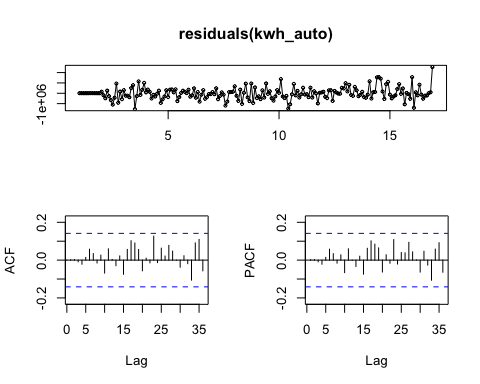


With the data relatively clean, we'll use auto.arima which suggests an ARIMA(1,0,1)(2,1,0) model. The residuals plots look reasonable as does the forecast plot with peaks in the winter and summer months and troughs in the spring and fall months.

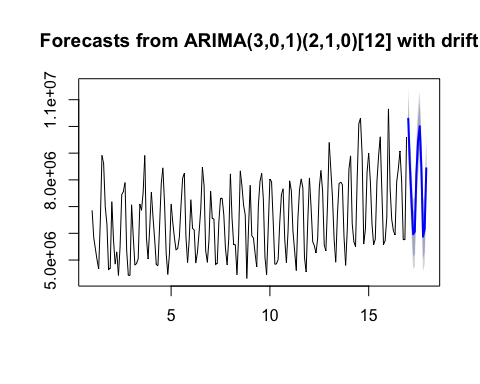
kwh\_auto = auto.arima(kwh\_ts)  
summary(kwh\_auto)

## Series: kwh\_ts   
## ARIMA(3,0,1)(2,1,0)[12] with drift   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1 sar1 sar2 drift  
## -0.3678 0.1432 0.1988 0.6912 -0.6976 -0.4153 9025.989  
## s.e. 0.2200 0.1078 0.0888 0.2208 0.0778 0.0788 3070.034  
##   
## sigma^2 estimated as 3.798e+11: log likelihood=-2655.61  
## AIC=5327.23 AICc=5328.07 BIC=5352.77  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -8097.783 584960.6 424585.7 -0.7902071 6.472204 0.6975321  
## ACF1  
## Training set 0.002244452

tsdisplay(residuals(kwh\_auto))



kwh\_forecast = forecast(kwh\_auto, h = 12)  
plot(kwh\_forecast)

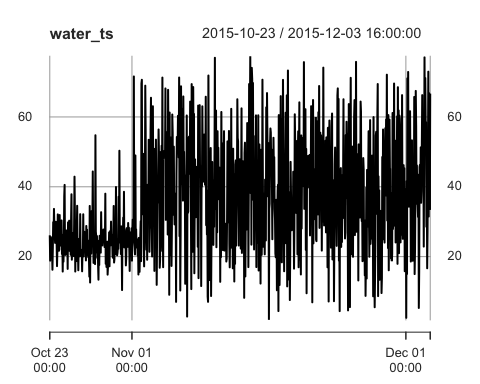


write.csv(kwh\_forecast$mean, '/Users/brucehao/Google Drive/CUNY/git/DATA624/Project 1/kwh\_forecast.csv')

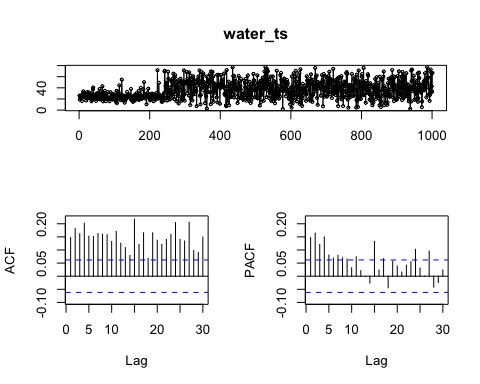
## Project 1: BONUS

First, we take the combined spreadsheets, group by day/hour and take the average flow per hour. Then, we apply the same plotting, auto.arima and forecasting methods as above.

water = read.csv('/Users/brucehao/Google Drive/CUNY/git/DATA624/Project 1/Waterflow\_Pipe.csv', stringsAsFactors = F)  
water\_agg = water %>%   
 mutate(dt = mdy\_hm(water$Date.Time)) %>%   
 mutate(dt = ymd\_h(paste(year(dt), month(dt), day(dt), hour(dt), sep = ' '))) %>%   
 group\_by(dt) %>%   
 summarise(AvgFlow = mean(WaterFlow))  
  
water\_ts = as.xts(water\_agg$AvgFlow, order.by = water\_agg$dt)  
attr(water\_ts, 'frequency') = 24  
  
plot(water\_ts)



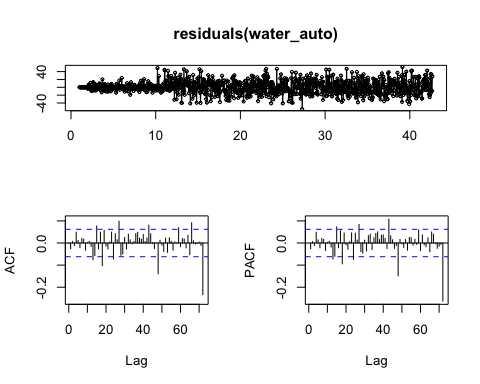
tsdisplay(water\_ts)



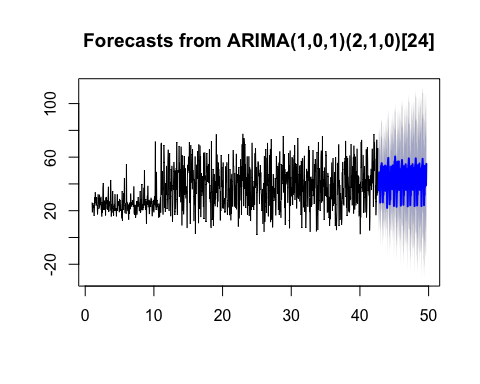
water\_auto = auto.arima(water\_ts, D = 1) #, stepwise = F, approximation = F)  
summary(water\_auto)

## Series: water\_ts   
## ARIMA(1,0,1)(2,1,0)[24]   
##   
## Coefficients:  
## ar1 ma1 sar1 sar2  
## 0.8732 -0.8252 -0.5753 -0.2755  
## s.e. 0.0860 0.0992 0.0312 0.0314  
##   
## sigma^2 estimated as 285: log likelihood=-4150.22  
## AIC=8310.43 AICc=8310.49 BIC=8334.86  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 0.6039733 16.64509 12.73528 -25.31283 51.54674 0.8423023  
## ACF1  
## Training set -0.02517306

tsdisplay(residuals(water\_auto))



water\_forecast = forecast(water\_auto, h = 7 \* 24)  
plot(water\_forecast)



write.csv(water\_forecast$mean, '/Users/brucehao/Google Drive/CUNY/git/DATA624/Project 1/water\_forecast.csv')