DATA 624: Project 1

Juliann McEachern

October 22, 2019

Contents

Overview

I am leaving the project overview page here for us to compile our final report in one singular document. We will add additional information here regarding project one to include explanation of process, etc.

Dependencies

Please add all libraries used here.

The following R libraries were used to complete Project 1:

```r
# General
library('easypackages')

libraries('knitr', 'kableExtra', 'default')

# Processing
libraries('readxl', 'tidyverse', 'janitor', 'lubridate')

# Graphing
libraries('ggplot2', 'grid', 'gridExtra', 'ggfortify','ggpubr')

# Timeseries
libraries('zoo', 'urca', 'tseries', 'timetk')

# Math
libraries('forecast')
```

Data

Data was stored within our group repository and imported below using the `readxl` package. Each individual question was solved within an R script and the data was sourced into our main report for discussion purposes. The R scripts are available within our appendix for replication purposes.

For grading purposes, we exported and saved all forecasts as a csv in our data folder.

```r
# Data Aquisition
atm_data <- read_excel("data/ATM624Data.xlsx")
power_data <- read_excel("data/ResidentialCustomerForecastLoad-624.xlsx")
pipe1_data <- read_excel("data/Waterflow_Pipe1.xlsx")
pipe2_data <- read_excel("data/Waterflow_Pipe2.xlsx")

# Source Code
source("scripts/Part-A-JM.R")
```

1 Part A

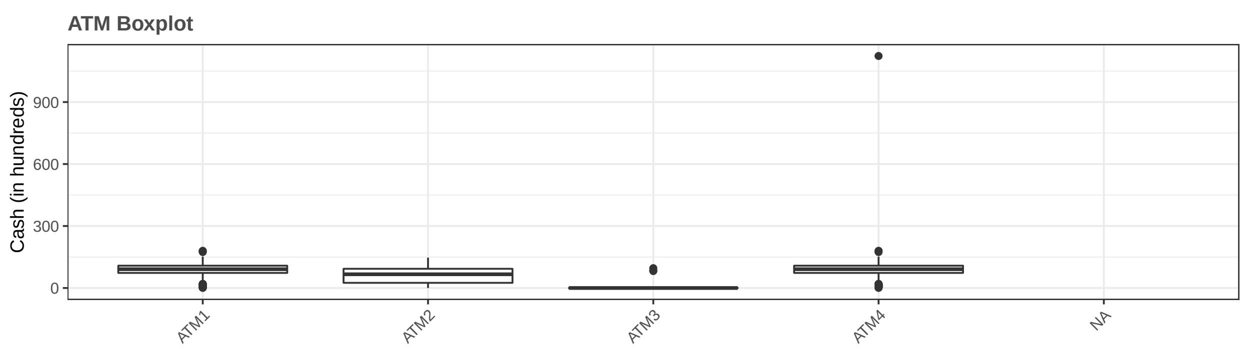
Instructions: In part A, I want you to forecast how much cash is taken out of 4 different ATM machines for May 2010. The data is given in a single file. The variable `Cash` is provided in hundreds of dollars, other than that it is straight forward. I am being somewhat ambiguous on purpose. I am giving you data, please provide your written report on your findings, visuals, discussion and your R code all within a Word readable document, except the forecast which you will put in an Excel readable file. I must be able to cut and paste your R code and run it in R studio. Your report must be professional - most of all - readable, EASY to follow. Let me know what you are thinking, assumptions you are making! Your forecast is a simple CSV or Excel file that MATCHES the format of the data I provide.

1.1 Exploration

The data covers a period of Friday May 1, 2010 through Saturday April 30, 2010. A forecast for the month of May will be 31 days in length.

While reviewing the data, we identified that the original data file contained `NA` values in our `ATM` and `Cash` columns for 14 observations between May 1 and 14, 2010. As these contain no information, we removed these missing values and transformed the dataset into a wide format.

We examined summary statistics for each ATM time series: * ATM1 and ATM2 have pretty normal distributions; ATM1's daily mean cash dispensed is $84, and ATM2's is $62. * ATM3 only dispensed cash on the last three days of the time series - as this provides few data points on which to forecast, we'll need to treat it specially. * ATM4 has a similar mean to ATM1, but skew and kurtosis suggest the impact of an outlier Wednesday, February 10, 2010. If this ATM is located in the Northeastern United States, this may have a relationship to a blizzard which struck on that day.



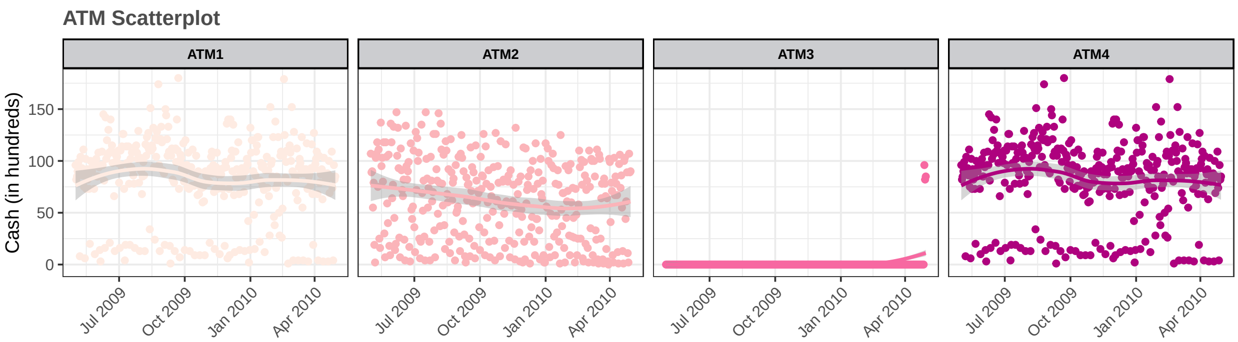

```
FALSE # A tibble: 3 x 2
FALSE   DATE                  Cash
FALSE   <dttm>               <dbl>
FALSE 1 2010-04-28 00:00:00     96
FALSE 2 2010-04-29 00:00:00     82
FALSE 3 2010-04-30 00:00:00     85
```

Our cleaned dataframe was then converted into a timeseries format using the `zoo` package for forecasting in the next section. Our initial review of the data showed that ATM2 contained one missing value on October 25, 2019 and that ATM4 contained a potential outlier of $1,123 on 2010-02-09. We replaced both values with the corresponding mean value of each machine.

Table 1.1: ATM Summary Statistics

group1	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	
ATM1	365	84.1013699	36.604256	91.0	86.86348	25.2042	1	180	179	-0.7186635	0.2087397	1.91595
ATM2	364	62.4642857	38.901108	66.5	62.08904	49.6671	0	147	147	-0.0268252	-1.0988678	2.03897
ATM3	365	0.7205479	7.944778	0.0	0.00000	0.0000	0	96	96	10.9291078	118.3807595	0.41584
ATM4	365	86.8410959	65.523479	91.0	86.86348	25.2042	1	1123	1122	10.6692960	168.6630832	3.42965

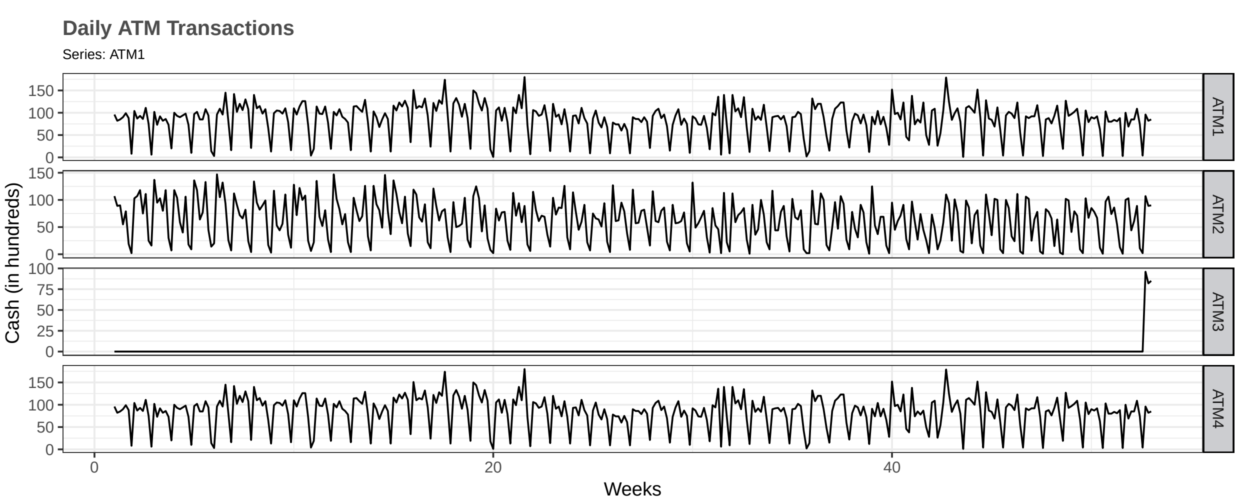
Next, we used a scatterplot to examine the correlation between cash withdrawals and dates for each machine. We identified similiar patterns between ATM1 and ATM4, which show non-linear fluctuations that suggest a potential trend component in these timeseries. ATM2 follows a relatively linear path and decreases overtime. This changes in the last few observations, where withdrawals begin to increase. There are only 3 observed transactions for ATM3 that appear at the end of the captured time period.



1.2 Timeseries Plots

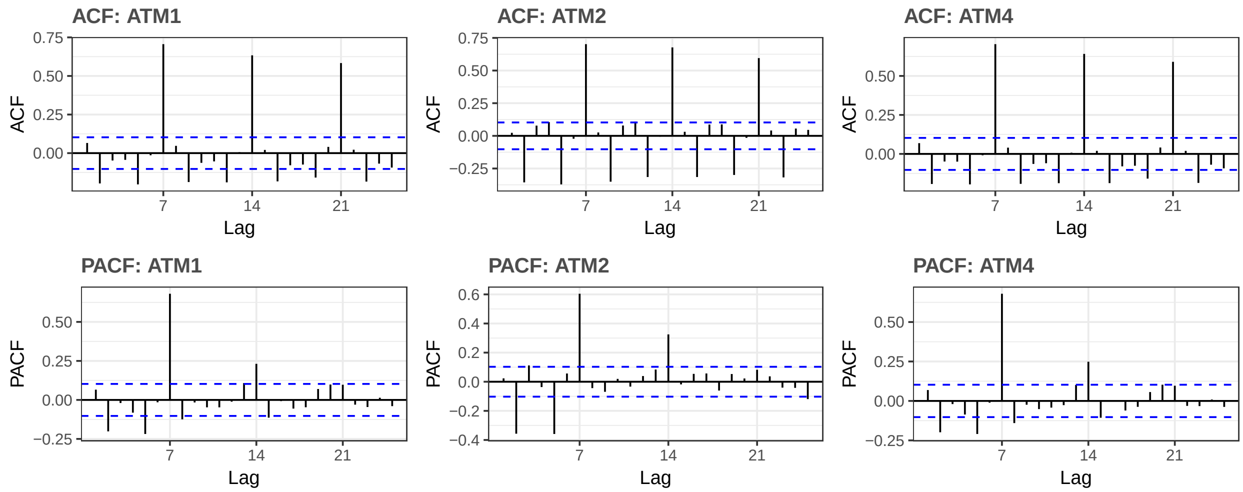
The time series plots show high weekly variance, for ATM1, ATM2, and ATM4 - consistent with our takeaway from the scatterplots.

These plots also remind us that ATM3 only dispensed cash on 3 days at the end of the timespan, with a daily range between $82 and $96. Given the paucity of observations in the training data, the simplest possible approach to forecasting ATM3 - averaging - is likely best. Given that ATM3 distributed no cash until April 28, 2010, we'll assume that it was not operating until then and only include the three day window of non-zero observations in the forecast.



1.3 Evaluation

We constructed our initial timeseries for ATM1, ATM2, and ATM4 using a weekly frequency. Our ACF plots for each ATM showcases large, decreasing lags starting at 7. This pattern continues in a multiple of seven, which confirms our assumption about seasonality within the observed data. These lags are indicative of a weekly pattern.



Our plots further suggest that the ATM data is non-stationary. We performed a unit root test using the `ur.kpss()` function to confirm this observation. The test results below show that differencing is required on all ATM2 and ATM4 series. ATM1 falls just below the cut-off critical value, but could still use differencing due to the observed seasonal pattern.

Table 1.2: KPSS unit root test

ATM	No-Diff	Diff-1
ATM1	0.4967	0.0219
ATM2	2.0006	0.016
ATM4	0.5182	0.0211

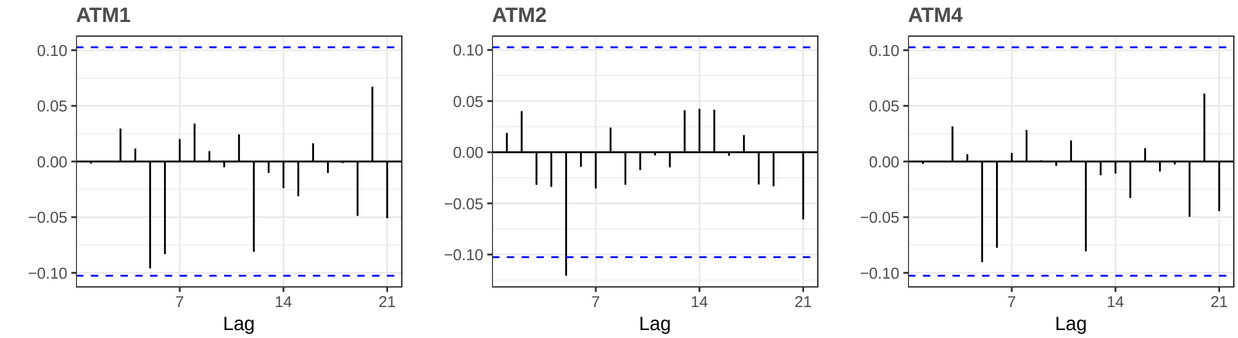
1.3.1 Modeling

We used `auto.arima()` and set `D=1` to account for seasonal differencing of our data to select the best ARIMA models for ATM1, ATM2, and ATM4. The full models and accuracy statistics for each series can be viewed in the appendix.

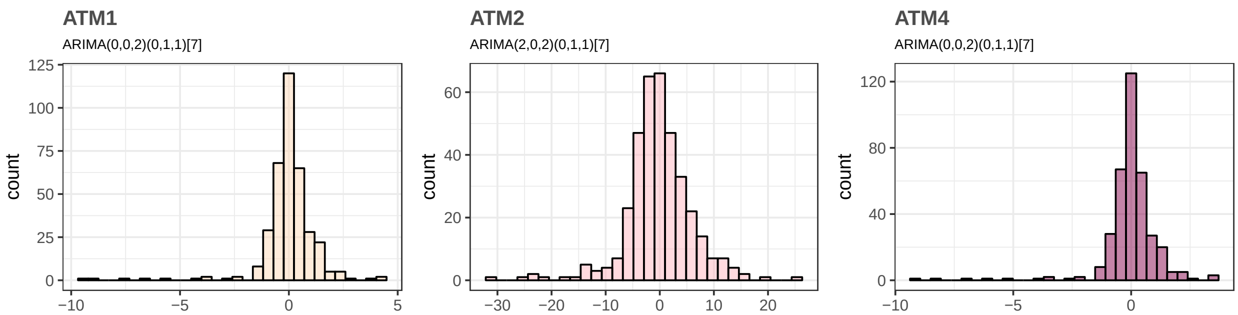
- **ATM1**: ARIMA$(0, 0, 2)(0, 1, 1)_7$
- **ATM2**: ARIMA$(2, 0, 2)(0, 1, 1)_7$
- **ATM3**: MEAN
- **ATM4**: ARIMA$(0, 0, 2)(0, 1, 1)_7$

The following ACF plots show us that our differentiated data is now stationary. Further, the residual histograms follow a relatively normal distribution, which confirms that the models adequately fits the observed data.

ACF Plots



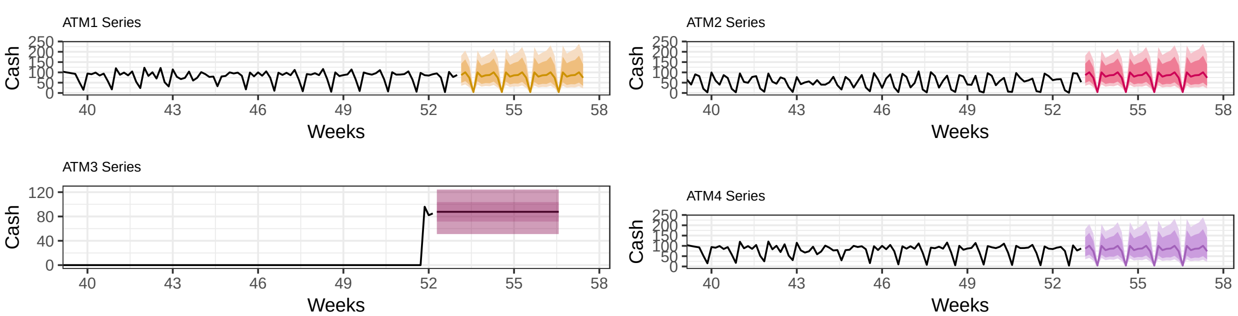
Residuals Plots



1.4 Forecast

Finally, we applied a forecast to each series for 31 days, which span across 5 weeks, in May 2010. The numeric forecasts can be viewed in a table output in the appendix section and are also located within our data output folder.

ATM Forecasts



Appendix

Part A

ARIMA Model Summary

ATM1:

```
FALSE Series: ATM1_ts
FALSE ARIMA(0,0,2)(0,1,1)[7]
FALSE Box Cox transformation: lambda= 0.2584338
FALSE
FALSE Coefficients:
FALSE          ma1      ma2     sma1
FALSE       0.1085  -0.1089  -0.6425
FALSE s.e.  0.0524   0.0521   0.0431
FALSE
FALSE sigma^2 estimated as 1.726:  log likelihood=-606.1
FALSE AIC=1220.2   AICc=1220.32   BIC=1235.72
```

ATM2:

```
FALSE Series: ATM2_ts
FALSE ARIMA(2,0,2)(0,1,1)[7]
FALSE Box Cox transformation: lambda= 0.661752
FALSE
FALSE Coefficients:
FALSE           ar1      ar2     ma1     ma2     sma1
FALSE       -0.4238  -0.8978  0.4766  0.7875  -0.7064
FALSE s.e.   0.0592   0.0473  0.0883  0.0608   0.0417
FALSE
FALSE sigma^2 estimated as 38.94:  log likelihood=-1162.96
FALSE AIC=2337.93   AICc=2338.17   BIC=2361.21
```

ATM3:

```
FALSE Mean of non-zero values is 87.67
```

ATM4:

```
FALSE Series: ATM4_ts
FALSE ARIMA(0,0,2)(0,1,1)[7]
FALSE Box Cox transformation: lambda= 0.2328582
FALSE
FALSE Coefficients:
FALSE          ma1      ma2     sma1
FALSE       0.1095  -0.1088  -0.6474
```

```
FALSE s.e.  0.0524   0.0523   0.0420
FALSE
FALSE sigma^2 estimated as 1.439:  log likelihood=-573.5
FALSE AIC=1154.99   AICc=1155.11   BIC=1170.52
```

Point Forecasts

Table 1.3: ATM Mean Point Forecast

Date	ATM1	ATM2	ATM3	ATM4
2010-05-01	86.6822230334281	65.9130078295388	87.6666666666667	86.7148793513953
2010-05-02	100.569237833094	71.2678744758685	87.6666666666667	100.581688969852
2010-05-03	73.710292000956	11.4694658108709	87.6666666666667	73.645362194735
2010-05-04	4.22902938718943	2.46415237936744	87.6666666666667	4.22143290375228
2010-05-05	100.159253208527	98.3397063045857	87.6666666666667	100.159422079885
2010-05-06	79.3467329167084	89.0607216505235	87.6666666666667	79.3417211032809
2010-05-07	85.7390398914344	66.0684601266365	87.6666666666667	85.7781984478913
2010-05-08	87.1797624007239	65.9067170158147	87.6666666666667	87.218337497544
2010-05-09	100.388112929695	71.3008782172556	87.6666666666667	100.395109536387
2010-05-10	73.710292000956	11.4650527078295	87.6666666666667	73.645362194735
2010-05-11	4.22902938718943	2.45577466621943	87.6666666666667	4.22143290375228
2010-05-12	100.159253208527	98.3602657501897	87.6666666666667	100.159422079885
2010-05-13	79.3467329167084	89.0776223553505	87.6666666666667	79.3417211032809
2010-05-14	85.7390398914344	66.0458523021009	87.6666666666667	85.7781984478913
2010-05-15	87.1797624007239	65.9025868757576	87.6666666666667	87.218337497544
2010-05-16	100.388112929695	71.3235063276986	87.6666666666667	100.395109536387
2010-05-17	73.710292000956	11.4619371370152	87.6666666666667	73.645362194735
2010-05-18	4.22902938718943	2.45005983249313	87.6666666666667	4.22143290375228
2010-05-19	100.159253208527	98.3744953596224	87.6666666666667	100.159422079885
2010-05-20	79.3467329167084	89.0890848253535	87.6666666666667	79.3417211032809
2010-05-21	85.7390398914344	66.0302973545638	87.6666666666667	85.7781984478913
2010-05-22	87.1797624007239	65.8998808213307	87.6666666666667	87.218337497544
2010-05-23	100.388112929695	71.3390190248587	87.6666666666667	100.395109536387
2010-05-24	73.710292000956	11.4597394828691	87.6666666666667	73.645362194735
2010-05-25	4.22902938718943	2.44616060460193	87.6666666666667	4.22143290375228
2010-05-26	100.159253208527	98.3843423320737	87.6666666666667	100.159422079885
2010-05-27	79.3467329167084	89.0968573387821	87.6666666666667	79.3417211032809
2010-05-28	85.7390398914344	66.0195955091203	87.6666666666667	85.7781984478913
2010-05-29	87.1797624007239	65.8981117895746	87.6666666666667	87.218337497544
2010-05-30	100.388112929695	71.3496527614802	87.6666666666667	100.395109536387
2010-05-31	73.710292000956	11.4581905724258	87.6666666666667	73.645362194735

R Script

?, commented out the previous write_csv operation and added in another, let's discuss

```
# load data
atm_data <- read_excel("data/ATM624Data.xlsx")

# clean dataframe
atm <- atm_data %>%
  # create wide dataframe
  spread(ATM, Cash) %>%
  # remove NA column using function from janitor package
  remove_empty(which = "cols") %>%
  # filter unobserved values from May 2010
  filter(DATE < as.Date("2010-05-01")) %>%
  # ensure dates are ascending
  arrange(DATE)

atm$ATM2[is.na(atm$ATM2)] <- mean(atm$ATM2, na.rm = TRUE) ## remove NA
atm$ATM4[which.max(atm$ATM4)] <- mean(atm$ATM4, na.rm = TRUE) ## remove outlier

# create TS with weekly frequency & subset data
atm_ts <- atm %>% select(-DATE) %>% ts(start=1,  frequency = 7)
ATM1_ts <- atm_ts[,1]; ATM2_ts <- atm_ts[,2]; ATM3_ts <- atm_ts[,3]; ATM4_ts <- atm_ts[,4]

#unit root test
## no diff
ATM1_ur <-ur.kpss(ATM1_ts)
ATM2_ur <-ur.kpss(ATM2_ts)
ATM4_ur <-ur.kpss(ATM4_ts)
## first order diff
ATM1d_ur <-ur.kpss(diff(ATM1_ts, lag=7))
ATM2d_ur <-ur.kpss(diff(ATM2_ts, lag=7))
ATM4d_ur <-ur.kpss(diff(ATM4_ts, lag=7))

# AUTO.ARIMA function; set D=1 for seasonal differencing
ATM1_AA <-auto.arima(ATM1_ts, D = 1, lambda = "auto", approximation = F, stepwise = T)
ATM2_AA <-auto.arima(ATM2_ts, D = 1, lambda = "auto", approximation = F, stepwise = T)
ATM4_AA <-auto.arima(ATM4_ts, D = 1, lambda = "auto", approximation = F, stepwise = T)

# Forecast Results
ATM1_fc <- forecast(ATM1_AA,h=31)
ATM2_fc <- forecast(ATM2_AA,h=31)
ATM3_fc <- meanf(ATM3_ts[ATM3_ts > 0], h=31)# based on three non-zero values (between observations 363
ATM4_fc <- forecast(ATM4_AA,h=31)

# Prepare dataframe for ATM3 mean forcast plotting
ATM3_plotdata_fc <- cbind(seq(from = 366, to = 396),
                          ATM3_fc[[5]],
                          ATM3_fc[[6]],
                          ATM3_fc[[7]]) %>%
                           as.data.frame()
```

```r
colnames(ATM3_plotdata_fc) <- c('Date', 'Point Forecast', 'Lo 80', 'Lo 95', 'Hi 80', 'Hi 95')
ATM3_plotdata <- ATM3_ts %>%
  fortify() %>%
  select(-Index) %>%
  rename(Cash = Data) %>%
  mutate(Date = as.numeric(row.names(.))) %>%
  select(Date, Cash) %>%
  full_join(ATM3_plotdata_fc, by = 'Date')

# Revert results back into original form
# date <- as.character(seq(as.Date('2010-05-01'), length.out=31, by=1))
# ATM_FC <-  cbind("Date"=date, "ATM1"=ATM1_fc$mean, "ATM2"=ATM2_fc$mean,
                 # "ATM3"=c(NA,NA,NA,NA),"ATM4"=ATM4_fc$mean) %>% as.data.frame()

# Combine the forecasts for the different ATMS
ATM_ALL_FC <- bind_cols(as.data.frame(ATM1_fc[4:6]),
                    as.data.frame(ATM2_fc[4:6]),
                    as.data.frame(ATM3_fc[5:7]),
                    as.data.frame(ATM4_fc[4:6])) %>%
  rename(ATM1_mean = 'mean',
         ATM1_low80CI = 'lower.80.',
         ATM1_low95CI = 'lower.95.',
         ATM1_upper80CI = 'upper.80.',
         ATM1_upper95CI = 'upper.95.',
         ATM2_mean = 'mean1',
         ATM2_low80CI = 'lower.80.1',
         ATM2_low95CI = 'lower.95.1',
         ATM2_upper80CI = 'upper.80.1',
         ATM2_upper95CI = 'upper.95.1',
         ATM3_mean = 'mean2',
         ATM3_low80CI = 'lower.80.2',
         ATM3_low95CI = 'lower.95.2',
         ATM3_upper80CI = 'upper.80.2',
         ATM3_upper95CI = 'upper.95.2',
         ATM4_mean = 'mean3',
         ATM4_low80CI = 'lower.80.3',
         ATM4_low95CI = 'lower.95.3',
         ATM4_upper80CI = 'upper.80.3',
         ATM4_upper95CI = 'upper.95.3'
         )

# Save output
write.csv(ATM_ALL_FC, file="forecasts/ATM_FC.csv")
```