DATA624-Project1

Silma

2025-03-24

DATA 624 - Predective Analytics

Project 1

Silma Khan

Spring 2025 As the industries continue to utilize data for deriving business insights, forecasting has become a crutial addition in aiding business initiatives and planning. This project shows a complete design and workflow for time series forecasting using two different scenarios. We will be covering data import, cleaning, exploration, analysis, model fitting, forecasting, and exporting our results. For this project, it is divided into two parts:

- **Part A - Forecasting ATM Cash Withdrawls**

In this part, we were provided an excel file titled `ATM624Data.xlsx` where we can analyze daily cash withdrawls from 4 different ATM machines. This file contains columns:

1. **DATE**: Where the information if provided in a date and timestamp, which we will convert into a normal date format (maybe create another column for time stamp)
2. **ATM**: where this is a categorical feature than indicates which ATM machine the withdrawl is pulling from. There are `ATM1`, `ATM2`, `ATM3`, and `ATM4`
3. **CASH**: and this contains the information regarding the amount of cash that is withdrawn on that date and from that row specific atm machine. Important to note, the cash is expressed in **hundreds of dollars**, so for example if the column contains 55, then the actual amount withdrawn is $5,500

The goal of Part A is to create a forecast of the daily cash withdrawls for each ATM machine for the entire month of May 2010, as the dates provided stop at April 2010. This will require transforming the data into a time series format where we will have one series per each ATM machine which will then allow us to forecast using a model and then predict the daily withdrawl amounts for the month of May 2010.

- **Part B - Forecasting Residential Power Usage**

In this part, we were provided with another excel file titled `ResidentialCustomerForecastLoad-624.xlsx` where we will now be using the monthly residential power usage data for years 1998-2013. This file contains the columns:

1. **CaseSequence**: which I believe is used as an identifier which is not necessary to use in the forecast
2. **YYY-MMM**: where this column contails the year and month in the format of 1998-Jan. This will need to go through some transformations and converting it into a standard date format

3. **KWH**: this column contains the power consumption for that month and year in the power unit kilowatt/hours

The goal of Part B is to forecast the power consumption for the months of 2014 since this dataset showcases data from Jan 1998 - Dec 2013. To do this, we would need to convert the date format into a standard date object, create a monthly time series, fitting a forecasting model, and finally generate the forecast for 2014.

First we need to load in our packages we need to perform this project:

```r
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.4.3
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.4.3

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library(openxlsx)
```

```
## Warning: package 'openxlsx' was built under R version 4.4.3
```

```
library(ggplot2)
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.4.3

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

Part 1 - ATM Cash Forcast

Data Import

```
getwd()
```

```
## [1] "C:/Users/khans/Downloads"
```

```
atm_data <- read_excel("ATM624Data.xlsx")

head(atm_data)
```

```
## # A tibble: 6 x 3
##    DATE ATM    Cash
##   <dbl> <chr> <dbl>
## 1 39934 ATM1     96
## 2 39934 ATM2    107
## 3 39935 ATM1     82
## 4 39935 ATM2     89
## 5 39936 ATM1     85
## 6 39936 ATM2     90
```

Data Preprocessing

```
str(atm_data$DATE)
```

```
##  num [1:1474] 39934 39934 39935 39935 39936 ...
```

```
atm_data$DATE <- as.Date(atm_data$DATE, origin = "1899-12-30")

head(atm_data$DATE)
```

```
## [1] "2009-05-01" "2009-05-01" "2009-05-02" "2009-05-02" "2009-05-03"
## [6] "2009-05-03"
```

```
head(atm_data)
```

```
## # A tibble: 6 x 3
##   DATE       ATM    Cash
##   <date>     <chr> <dbl>
## 1 2009-05-01 ATM1     96
## 2 2009-05-01 ATM2    107
## 3 2009-05-02 ATM1     82
## 4 2009-05-02 ATM2     89
## 5 2009-05-03 ATM1     85
## 6 2009-05-03 ATM2     90
```

After loading the data, we also converted the Excel dates to a proper date object for easier analysis and forecasting

Now, since the ATM machines are all included into one column, I think it would be better to create a wide table format to analyze each ATM component before doing that, we should also check to see if there are any rows with missing ATM values

```
unique(atm_data$ATM)
```

```
## [1] "ATM1" "ATM2" NA     "ATM3" "ATM4"
```

As we can see, NA is part of the count meaning there are some rows that have missing ATM information. To deal with this, we can simply remove rows with missing ATM values

```
atm_data_clean <- atm_data %>%
  filter(!is.na(ATM))
```

After cleaning the data, we can now move onto creating a wide format for the table

```
atm_wide <- atm_data_clean %>%
  select(DATE, ATM, Cash) %>%
  pivot_wider(names_from = ATM, values_from = Cash)

head(atm_wide)
```

```
## # A tibble: 6 x 5
##   DATE        ATM1  ATM2  ATM3  ATM4
##   <date>     <dbl> <dbl> <dbl> <dbl>
## 1 2009-05-01    96   107     0 777.
## 2 2009-05-02    82    89     0 524.
## 3 2009-05-03    85    90     0 793.
## 4 2009-05-04    90    55     0 908.
## 5 2009-05-05    99    79     0  52.8
## 6 2009-05-06    88    19     0  52.2
```

Now each row represents a single date and can see columns for the separate ATM machines with their respective cash values withdrawn

Since the date starts from May 2009 and we would like to forecast the cash withdraws from May 2010, we can convert each ATM's cash data into a daily time series

```
atm_wide <- atm_wide %>% arrange(DATE)

create_daily_ts <- function(x, start_date) {
  start_year <- year(start_date)
  start_doy <- yday(start_date)
  ts(x, start = c(start_year, start_doy), frequency = 365)
}

ts_ATM1 <- create_daily_ts(atm_wide$ATM1, min(atm_wide$DATE))
ts_ATM2 <- create_daily_ts(atm_wide$ATM2, min(atm_wide$DATE))
ts_ATM3 <- create_daily_ts(atm_wide$ATM3, min(atm_wide$DATE))
ts_ATM4 <- create_daily_ts(atm_wide$ATM4, min(atm_wide$DATE))
```

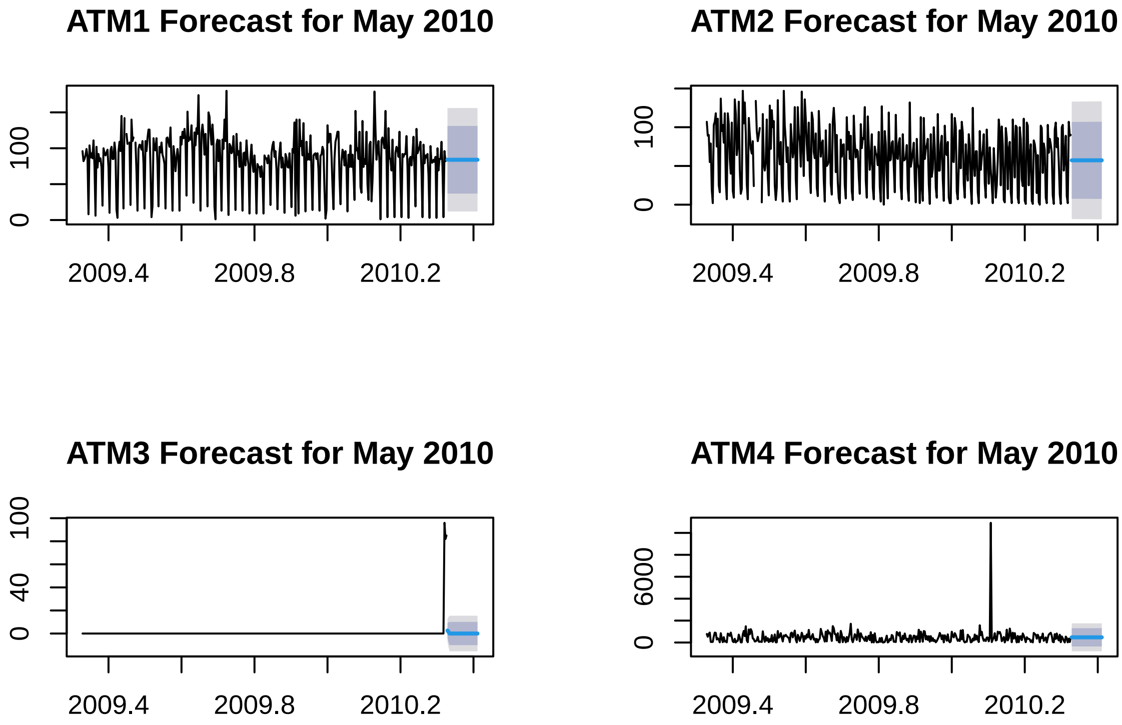
By creating a time series object for each ATM machine, we are able to help forecast the information for May 2010

```
tail(atm_wide)
```

```
## # A tibble: 6 x 5
##   DATE        ATM1  ATM2  ATM3  ATM4
##   <date>     <dbl> <dbl> <dbl> <dbl>
## 1 2010-04-25   109    89     0 542.
## 2 2010-04-26    74    11     0 404.
## 3 2010-04-27     4     2     0  13.7
## 4 2010-04-28    96   107    96 348.
## 5 2010-04-29    82    89    82  44.2
## 6 2010-04-30    85    90    85 482.
```

Taking a look at the data, we can see that the historical data ends on April 30, 2010 so we need to forecast for the 31 days in May from May 1, 2010 - May 31, 2010. To do this I will be using the ARIMA model for each series, and using the `auto.arima()` function to automatically select an appropriate ARIMA model

```
horizon <- 31

fit_and_forecast <- function(ts_data, h = horizon) {
  model <- auto.arima(ts_data)
  forecast(model, h = h)
}

fc_ATM1 <- fit_and_forecast(ts_ATM1)
fc_ATM2 <- fit_and_forecast(ts_ATM2)
fc_ATM3 <- fit_and_forecast(ts_ATM3)
fc_ATM4 <- fit_and_forecast(ts_ATM4)

par(mfrow = c(2, 2))
plot(fc_ATM1, main = "ATM1 Forecast for May 2010")
plot(fc_ATM2, main = "ATM2 Forecast for May 2010")
plot(fc_ATM3, main = "ATM3 Forecast for May 2010")
plot(fc_ATM4, main = "ATM4 Forecast for May 2010")
```


ATM1 Forecast for May 2010

ATM2 Forecast for May 2010

ATM3 Forecast for May 2010

ATM4 Forecast for May 2010

```r
par(mfrow = c(1, 1))
```

We fitted an ARIMA model to each ATM's time series and then forecast the next 31 days for the month of May.

Now we can compile the forecasted dailt value into a data frame and then export then to an excel file as the project outlines

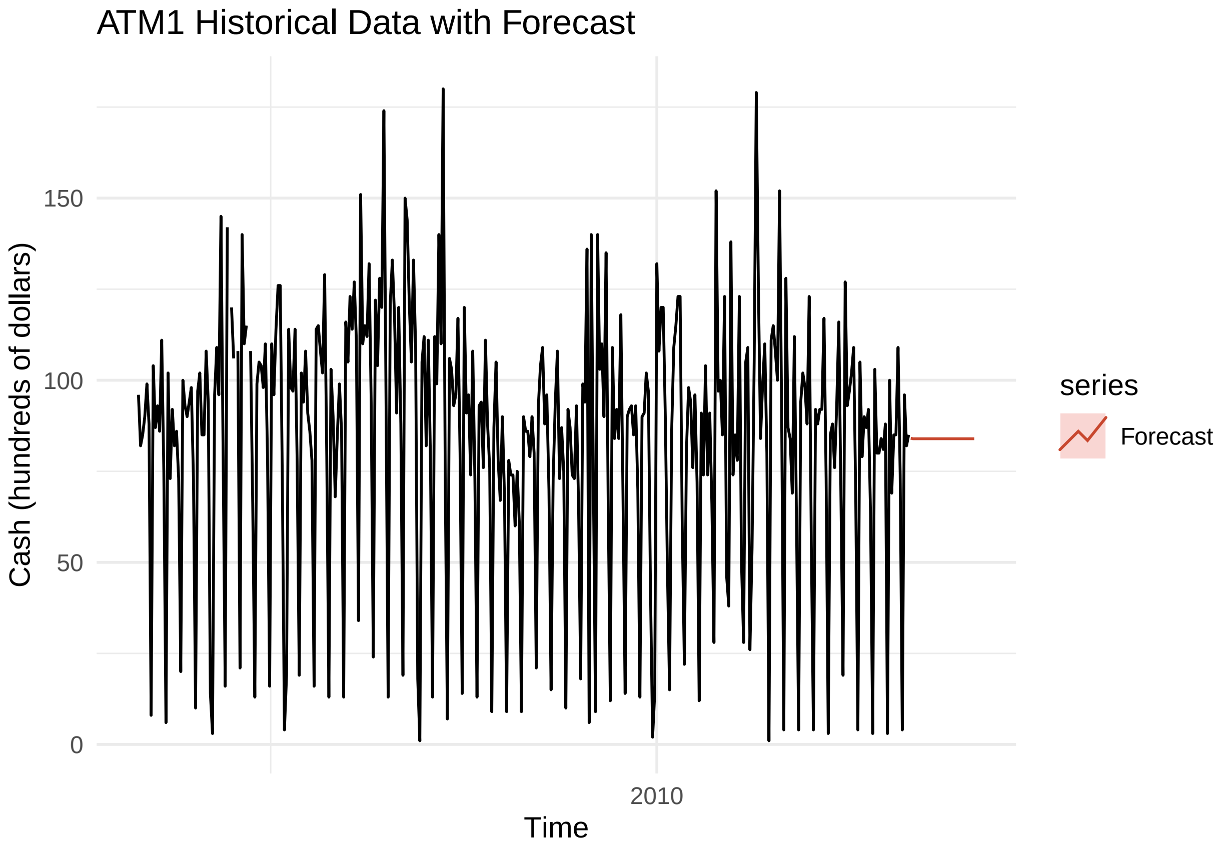
```r
end_history <- as.Date("2010-04-30")
forecast_dates <- seq.Date(end_history + 1, by = "day", length.out = horizon)

atm_forecasts <- data.frame(
  Date = forecast_dates,
  ATM1 = as.numeric(fc_ATM1$mean),
  ATM2 = as.numeric(fc_ATM2$mean),
  ATM3 = as.numeric(fc_ATM3$mean),
  ATM4 = as.numeric(fc_ATM4$mean)
)

head(atm_forecasts, 10)
```

```
##          Date     ATM1     ATM2     ATM3     ATM4
## 1  2010-05-01 84.07464 57.26867 2.610608 474.0433
## 2  2010-05-02 83.94107 57.26867 1.408144 474.0433
## 3  2010-05-03 83.94107 57.26867 0.000000 474.0433
## 4  2010-05-04 83.94107 57.26867 0.000000 474.0433
## 5  2010-05-05 83.94107 57.26867 0.000000 474.0433
## 6  2010-05-06 83.94107 57.26867 0.000000 474.0433
## 7  2010-05-07 83.94107 57.26867 0.000000 474.0433
## 8  2010-05-08 83.94107 57.26867 0.000000 474.0433
## 9  2010-05-09 83.94107 57.26867 0.000000 474.0433
## 10 2010-05-10 83.94107 57.26867 0.000000 474.0433
```

Now we can export the forecasts to an excel file

```
write.xlsx(atm_forecasts, file = "ATM_Forecast_May2010.xlsx", sheetName = "Forecast", rowNames = FALSE)
```

```
autoplot(ts_ATM1) +
  autolayer(fc_ATM1, series="Forecast", PI=FALSE) +
  labs(title="ATM1 Historical Data with Forecast", y="Cash (hundreds of dollars)", x="Time") +
  theme_minimal()
```



Part B - Residential Power FOrecast

```
power_data <- read_excel("ResidentialCustomerForecastLoad-624.xlsx")

head(power_data)
```

```
## # A tibble: 6 x 3
##   CaseSequence ‘YYYY-MMM‘     KWH
##          <dbl> <chr>        <dbl>
## 1          733 1998-Jan   6862583
## 2          734 1998-Feb   5838198
## 3          735 1998-Mar   5420658
## 4          736 1998-Apr   5010364
## 5          737 1998-May   4665377
## 6          738 1998-Jun   6467147
```

Taking a look at this data, we would need to convert the `YYYY-MMM` column into a proper year-month

```r
power_data <- power_data %>%
  mutate(`YYYY-MMM` = as.yearmon(`YYYY-MMM`, format = "%Y-%b"))

str(power_data$`YYYY-MMM`)
```

```
##  'yearmon' num [1:192] Jan 1998 Feb 1998 Mar 1998 Apr 1998 ...
```

```r
head(power_data$`YYYY-MMM`)
```

```
## [1] "Jan 1998" "Feb 1998" "Mar 1998" "Apr 1998" "May 1998" "Jun 1998"
```

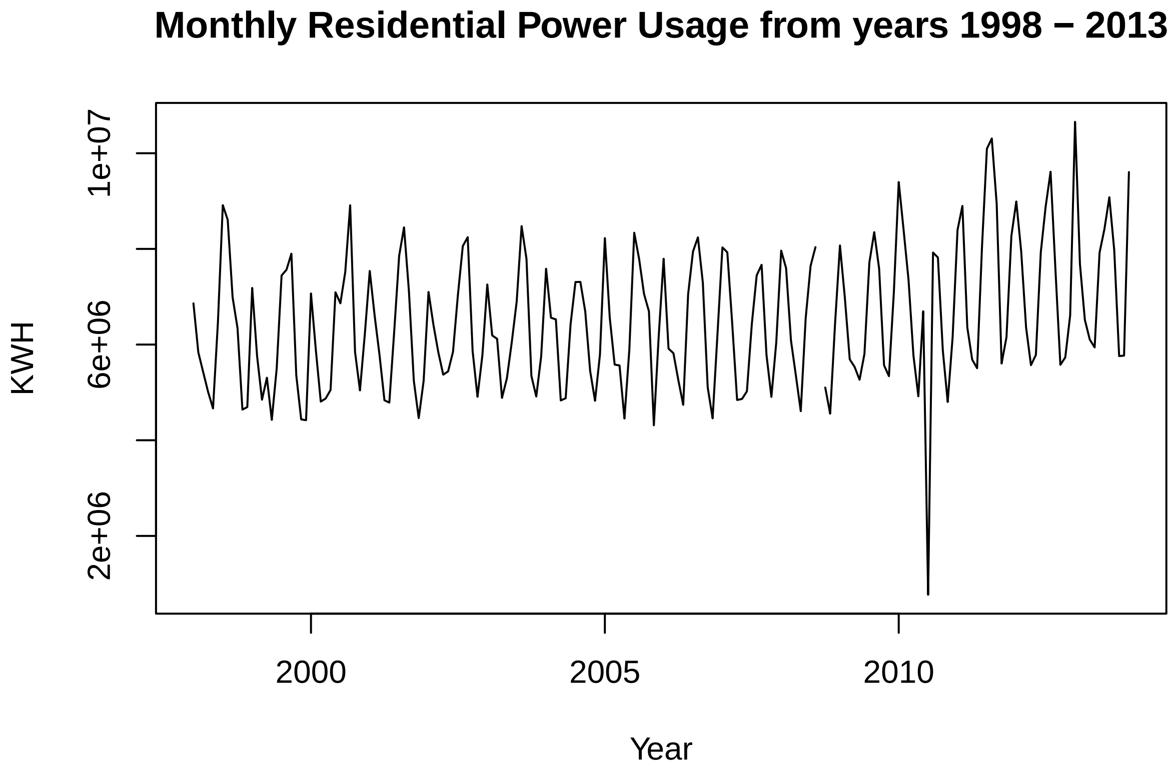
```r
head(power_data)
```

```
## # A tibble: 6 x 3
##   CaseSequence ‘YYYY-MMM‘     KWH
##          <dbl> <yearmon>    <dbl>
## 1          733 Jan 1998   6862583
## 2          734 Feb 1998   5838198
## 3          735 Mar 1998   5420658
## 4          736 Apr 1998   5010364
## 5          737 May 1998   4665377
## 6          738 Jun 1998   6467147
```

Now that we converted our time column into a friendlier format, we can move onto creating a monthly time series

Since the data spans from January 1998 to December 2013, we want to specify a frequency of 12 to indicate that we are dealing with monthly data

```r
ts_power <- ts(power_data$KWH, start = c(1998, 1), frequency = 12)

plot(ts_power, main = "Monthly Residential Power Usage from years 1998 - 2013",
     ylab = "KWH", xlab = "Year")
```

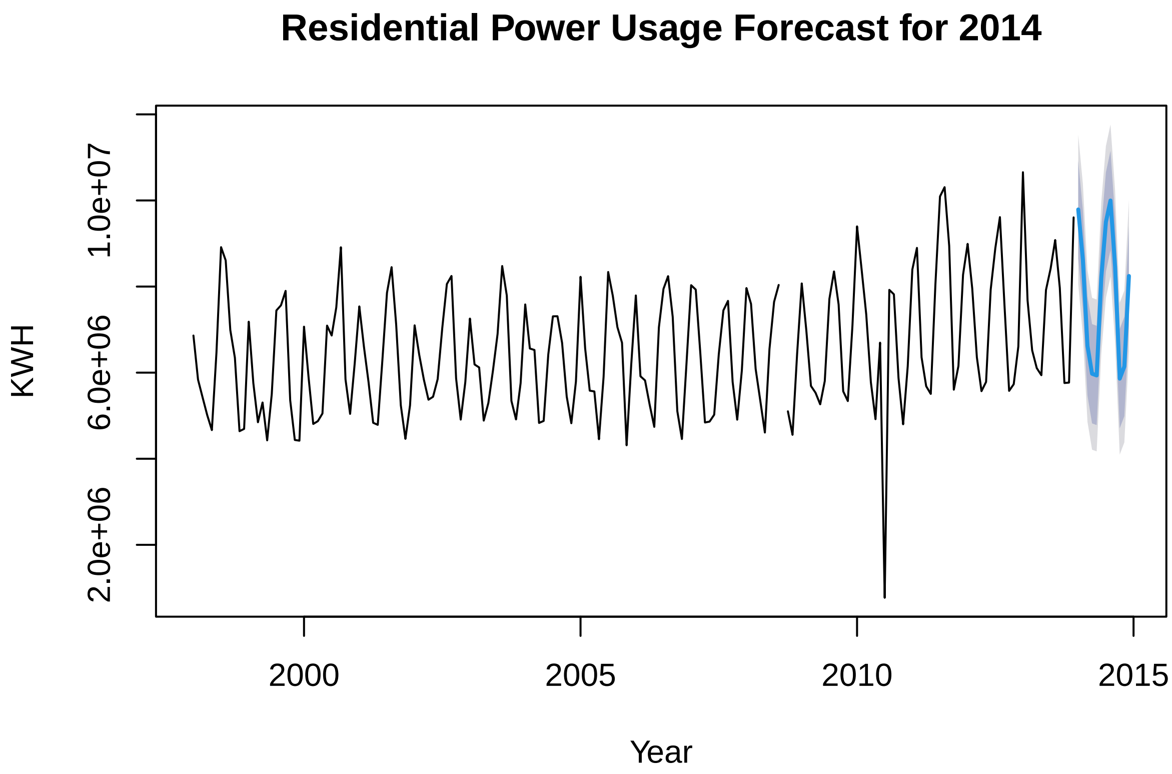


For this portion, for model fitting and forecasting the power consumption for 2014, we will also use the `auto.arima()` function to select the best ARIMA model for the historical data and then help us forecast the next 12 months for 2014

```
fit_power <- auto.arima(ts_power)
summary(fit_power)
```

```
## Series: ts_power
## ARIMA(0,0,2)(2,1,0)[12] with drift
##
## Coefficients:
##          ma1     ma2     sar1     sar2     drift
##       0.1739  0.0505  -0.7591  -0.4124  8750.907
## s.e.  0.0766  0.0844   0.0697   0.0682  3214.839
##
## sigma^2 = 7.841e+11:  log likelihood = -2707.12
## AIC=5426.25   AICc=5426.73   BIC=5445.4
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE
## Training set -9730.731 845160.6 506279.9 -5.055329 11.57781 0.730904
##                     ACF1
## Training set 0.008533477
```

```r
fc_power <- forecast(fit_power, h = 12)

plot(fc_power, main = "Residential Power Usage Forecast for 2014",
     ylab = "KWH", xlab = "Year")
```



We can see a general seasonality with the forecast of power comsumption for 2014 from the historical data and see some reflections of the sudden dip in poer consumption around 2010-2011

We can now compile the forecasted monthly values into a data frame. TO represent the forecasted months, we can generate a sequence of year mon objects for 2014 and then format them as strings to showcase

```r
forecast_months <- seq(as.yearmon("2014-01"), by = 1/12, length.out = 12)

power_forecast_df <- data.frame(
  Month = format(forecast_months, "%Y-%b"),
  Forecast = as.numeric(fc_power$mean)
)

head(power_forecast_df)
```

```
##      Month Forecast
## 1 2014-Jan  9790458
## 2 2014-Feb  8624753
## 3 2014-Mar  6623643
## 4 2014-Apr  5974459
## 5 2014-May  5935072
## 6 2014-Jun  8199158
```

We can also export the forecasts to an excel file

```
write.xlsx(power_forecast_df, file = "ResidentialPowerForecast_2014.xlsx", sheetName = "Forecast", rowNa
```

Conclusion

- **Part A - ATM Cash Forecasting**:

For this portion, we were tasked with importing and preprocessing daily ATM transaction data and converting/manipulating the excel file to forecast the cash transaction data for May 2010. We applied to `auto.arima()` function to generate forecasts. Although some ATM showed little seasonality and variability in the data, this may be due to some aspects of the historical data.

- **Part B - Residential Power Usage Forecasting**:

For this portion, we were given an excel sheet containing residential power usage data and we worked with monthly observations which spanned across Janurary 1998 - December 2013. We had to preprocess the data and manipulate the data to get a very time frame column. We created a monthly time series also using the `auto.arima()` method to fit the forecast. We also forecasted for 2014 power usage.

Overall, this project allowed for a comprehensive workflow for time series forecasting. Both aspects of this project portrayed the importance of understanding the data, what we are trying to achieve, and proper preprocessing steps. These forecasts can be useful for further analysis and aid in using other models as well.