DATA 624 Project 1

Warner Alexis

2025-03-30

## **Project 1**

This project consists of 3 parts - two required and one bonus and is worth 15% of your grade. The project is due at 11:59 PM on Sunday Oct 25. I will accept late submissions with a penalty until the meetup after that when we review some projects.

**ATM Forecast** The bar graph illustrates the total cash withdrawals (in hundreds of dollars) for each ATM machine across the dataset. ATM4 stands out with the highest total cash withdrawn at **173,026**, followed by ATM1 and ATM2 with **30,367** and **22,716** respectively. ATM3, on the other hand, shows a significantly lower total of just **263**, suggesting it was either inactive or only recently deployed during the data collection period. The underlying dataset contains **3 columns** (DATE, ATM, Cash) and **1,474 rows**, representing individual withdrawal records across different dates and machines. This analysis helps identify usage intensity and transaction volume per ATM, which is essential for forecasting and operational planning.

# Loading Libraries   
library(readxl)  
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

library(ggplot2)  
library(lubridate)

##   
## Attaching package: 'lubridate'

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

library(forecast)

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

library(tidyr)  
library(tsibble)

## Registered S3 method overwritten by 'tsibble':  
## method from   
## as\_tibble.grouped\_df dplyr

##   
## Attaching package: 'tsibble'

## The following object is masked from 'package:lubridate':  
##   
## interval

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

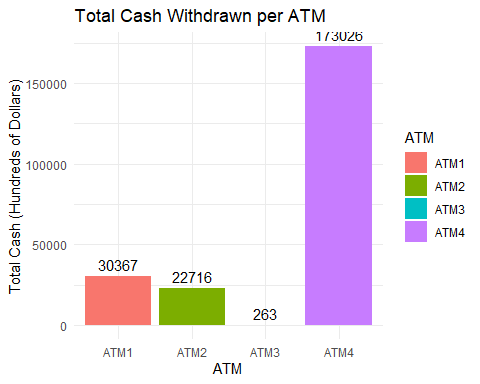
library(feasts)

## Loading required package: fabletools

library(fable)  
library(urca)  
library(writexl)  
  
df <- read\_excel("ATM624Data.xlsx", sheet = "ATM Data")  
  
# change df timestamps because of the way excell interpret it   
df <- df %>% mutate(DATE = as.Date(DATE, origin = "1899-12-30"))  
head(df)

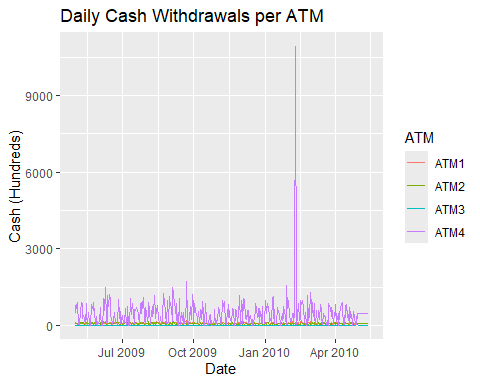
## # A tibble: 6 × 3  
## DATE ATM Cash  
## <date> <chr> <dbl>  
## 1 2009-05-01 ATM1 96  
## 2 2009-05-01 ATM2 107  
## 3 2009-05-02 ATM1 82  
## 4 2009-05-02 ATM2 89  
## 5 2009-05-03 ATM1 85  
## 6 2009-05-03 ATM2 90

# Summarize total cash per ATM with NA handled  
atm\_totals <- df |>  
 group\_by(ATM) |>  
 summarise(total\_cash = sum(Cash, na.rm = TRUE)) |>  
 filter(!is.na(ATM)) # Remove any NA ATM labels  
  
# Create bar plot  
ggplot(atm\_totals, aes(x = ATM, y = total\_cash, fill = ATM)) +  
 geom\_bar(stat = "identity") +  
 geom\_text(aes(label = round(total\_cash, 0)), vjust = -0.5) +  
 labs(title = "Total Cash Withdrawn per ATM",  
 x = "ATM",  
 y = "Total Cash (Hundreds of Dollars)") +  
 theme\_minimal()

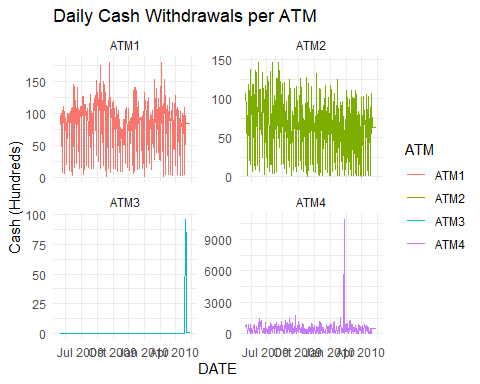


we first created a complete daily date sequence and merged it with our ATM withdrawal data to ensure no days are missing. Any missing values (i.e., days with no withdrawals) were filled with mean values for each ATM. Then, we reshaped the data from wide to long format and converted it into a tsibble, which is required for time series plotting using the autoplot() function. Finally, we generated a faceted time series plot with individual y-axis scales for each ATM, allowing us to clearly visualize daily withdrawal patterns across all machines.

# Ensure complete time series per ATM  
data\_wide <- df %>%  
 group\_by(ATM, DATE) %>%  
 summarise(Cash = sum(Cash), .groups = 'drop') %>%  
 tidyr::pivot\_wider(names\_from = ATM, values\_from = Cash)  
# Fill in missing days with 0s (if needed)  
full\_dates <- seq(min(df$DATE), max(df$DATE), by = "day")  
atm\_means <- data\_wide %>%  
 summarise(  
 ATM1 = mean(ATM1, na.rm = TRUE),  
 ATM2 = mean(ATM2, na.rm = TRUE),  
 ATM3 = mean(ATM3, na.rm = TRUE),  
 ATM4 = mean(ATM4, na.rm = TRUE)  
 )  
  
# Fill missing values with those means  
data\_full <- data.frame(DATE = full\_dates) %>%  
 left\_join(data\_wide, by = "DATE") %>%  
 tidyr::replace\_na(list(  
 ATM1 = atm\_means$ATM1,  
 ATM2 = atm\_means$ATM2,  
 ATM3 = atm\_means$ATM3,  
 ATM4 = atm\_means$ATM4  
 ))  
  
# Convert to long format and tsibble  
data\_long <- data\_full %>%  
 pivot\_longer(cols = starts\_with("ATM"), names\_to = "ATM", values\_to = "Cash") %>%  
 as\_tsibble(index = DATE, key = ATM)  
  
# Plot using autoplot  
autoplot(data\_long, Cash) +  
 labs(title = "Daily Cash Withdrawals per ATM", y = "Cash (Hundreds)", x = "Date")



# Plot with autoplot and free scales  
autoplot(data\_long, Cash) +  
 facet\_wrap(~ATM, scales = "free\_y") +  
 labs(title = "Daily Cash Withdrawals per ATM",  
 x = "DATE",  
 y = "Cash (Hundreds)") +  
 theme\_minimal()

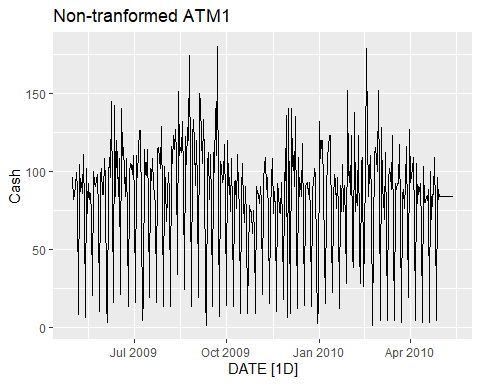


## ATM Forecast

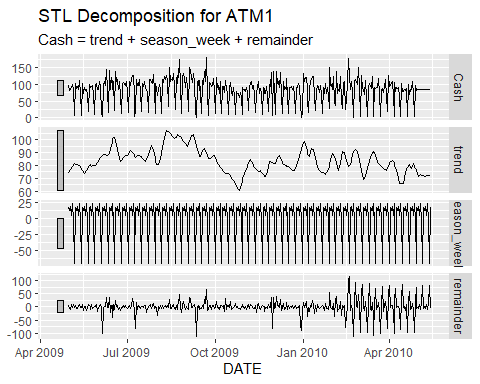
### ATM1

The Augmented Dickey-Fuller (ADF) test results indicate that the differenced time series is stationary. The test statistic (tau) is -23.331, which is far below the 1% critical value of -3.44. This suggests strong evidence against the null hypothesis of a unit root, meaning the time series does not have a unit root and is therefore stationary. Additionally, the p-value associated with the test is less than 0.01 (often reported as < 2.2e-16), which reinforces this conclusion. A low p-value (typically < 0.05) indicates that the observed data is highly inconsistent with the assumption of non-stationarity. Therefore, we reject the null hypothesis and conclude that the differenced series is stationary, making it suitable for time series modeling approaches that assume stationarity.

# Create tsibble  
ts\_data <- data\_long %>%  
 as\_tsibble(index = DATE, key = ATM)  
  
# for atm1   
  
ts\_data %>%  
 filter(ATM == "ATM1") %>%  
 autoplot(Cash) +  
 ggtitle("Non-tranformed ATM1")



ts\_data %>%  
 filter(ATM == "ATM1") %>%  
 model(STL(Cash ~ season(window = "periodic"), robust = TRUE)) %>%  
 components() %>%  
 autoplot() +  
 labs(title = "STL Decomposition for ATM1")



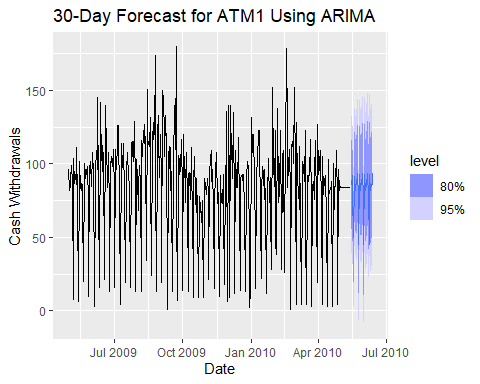
ts\_data <- ts\_data %>% select(-`NA`)  
# STL decomposition  
stl\_model <- ts\_data %>%  
 filter(ATM == "ATM1") %>%  
 model(STL(Cash ~ season(window = "periodic")))  
  
# Extract components  
components <- components(stl\_model)  
  
# Check if remainder is stationary (Augmented Dickey-Fuller test)  
  
adf\_test <- ur.df(components$remainder, type = "drift", selectlags = "AIC")  
summary(adf\_test)

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression drift   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -87.595 -9.177 0.089 12.061 76.842   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.01938 1.17240 -0.017 0.987   
## z.lag.1 -1.58670 0.06801 -23.331 <2e-16 \*\*\*  
## z.diff.lag 0.45778 0.04623 9.902 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 22.76 on 374 degrees of freedom  
## Multiple R-squared: 0.6389, Adjusted R-squared: 0.637   
## F-statistic: 330.9 on 2 and 374 DF, p-value: < 2.2e-16  
##   
##   
## Value of test-statistic is: -23.331 272.1682   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau2 -3.44 -2.87 -2.57  
## phi1 6.47 4.61 3.79

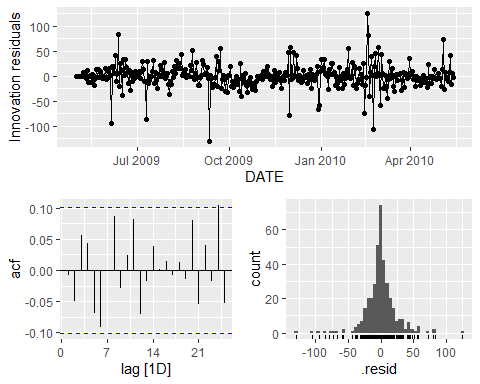
# Step 1: Filter data for ATM1  
atm1\_tsibble <- ts\_data %>%  
 filter(ATM == "ATM1")  
  
# Step 2: Fit ARIMA model (automatically selected orders)  
arima\_model <- atm1\_tsibble %>%  
 model(ARIMA(Cash))  
  
# Step 3: Model report  
report(arima\_model)

## Series: Cash   
## Model: ARIMA(0,0,1)(0,1,2)[7]   
##   
## Coefficients:  
## ma1 sma1 sma2  
## 0.1543 -0.5794 -0.1216  
## s.e. 0.0548 0.0500 0.0510  
##   
## sigma^2 estimated as 566.4: log likelihood=-1707.56  
## AIC=3423.12 AICc=3423.23 BIC=3438.8

# Step 4: Forecast 30 days ahead  
forecast\_arima <- arima\_model %>%  
 forecast(h = "30 days")  
  
# Step 5: Plot forecast with historical data  
autoplot(forecast\_arima, atm1\_tsibble) +  
 labs(title = "30-Day Forecast for ATM1 Using ARIMA",  
 y = "Cash Withdrawals", x = "Date")



# Step 6: Residual diagnostics  
# a) ACF plot of residuals  
arima\_model %>%  
 gg\_tsresiduals()



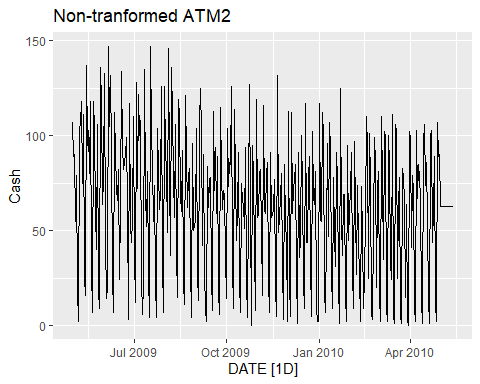
# b) Ljung-Box test for white noise residuals  
arima\_model %>%  
 augment() %>%  
 features(.innov, ljung\_box, lag = 24, dof = 3)

## # A tibble: 1 × 4  
## ATM .model lb\_stat lb\_pvalue  
## <chr> <chr> <dbl> <dbl>  
## 1 ATM1 ARIMA(Cash) 26.2 0.200

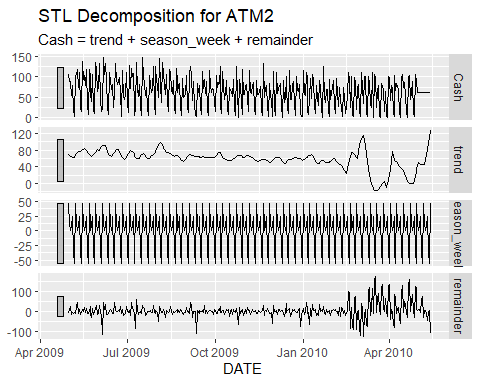
# extract point from forecast only   
forecast\_atm1 <- forecast\_arima %>%  
 as\_tibble() %>%  
 select(DATE, ATM, .mean) %>%  
 pivot\_wider(names\_from = ATM, values\_from = .mean)  
  
# Save to Excel  
write\_xlsx(forecast\_atm1, "ATM1\_Forecast\_May2010.xlsx")

### ATM2

ts\_data %>%  
 filter(ATM == "ATM2") %>%  
 autoplot(Cash) +  
 ggtitle("Non-tranformed ATM2")



ts\_data %>%  
 filter(ATM == "ATM2") %>%  
 model(STL(Cash ~ season(window = "periodic"), robust = TRUE)) %>%  
 components() %>%  
 autoplot() +  
 labs(title = "STL Decomposition for ATM2")



# --- 1. Extract ATM2 from data\_long ---  
atm2\_data <- ts\_data %>%  
 filter(ATM == "ATM2")  
  
# --- 2. Check if it's stationary (ADF test) ---  
atm2\_data %>% features(Cash, unitroot\_kpss)

## # A tibble: 1 × 3  
## ATM kpss\_stat kpss\_pvalue  
## <chr> <dbl> <dbl>  
## 1 ATM2 2.07 0.01

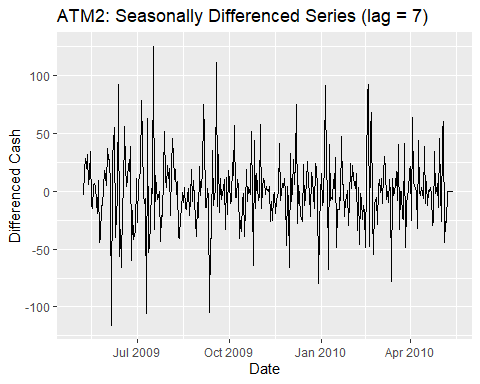
# --- 3. Differencing to remove trend and seasonality ---  
# First difference: removes trend  
# Seasonal difference: removes weekly seasonality (lag = 7)  
atm2\_stationary <- atm2\_data %>%  
 mutate(diff\_cash = difference(Cash, differences = 1),  
 diff\_seasonal = difference(Cash, lag = 7))

The null hypothesis of the KPSS test is that the series is stationary Since the p-value is 0.01 , we reject the null hypothesis. the unitroop\_kpps test confirms that the ATM2 series is non-stationary and requires differencing before modeling.

These plots help you visually inspect if: - Residuals are approximately normally distributed - There’s no autocorrelation left (ACF spikes should be within the blue bands) - The model is well-behaved (i.e., white noise residuals)

The Ljung-Box test was conducted on the residuals of the ARIMA model fitted to ATM2 in order to assess whether any autocorrelation remained after model fitting. The test returned a statistic of 8.47 with a p-value of 0.583 at lag 10. Since the p-value is greater than the conventional threshold of 0.05, we fail to reject the null hypothesis that the residuals are independently distributed. This indicates that the residuals behave like white noise and do not exhibit significant autocorrelation, suggesting that the ARIMA model for ATM2 provides an adequate fit to the data.

# Plot transformed (stationary) series  
atm2\_stationary %>%  
 autoplot(diff\_seasonal, na.rm = TRUE) +  
 labs(title = "ATM2: Seasonally Differenced Series (lag = 7)",  
 y = "Differenced Cash", x = "Date")



# Apply first-order differencing  
atm2\_diff <- atm2\_data %>%  
 mutate(diff\_cash = difference(Cash, differences = 1)) %>%  
 filter(!is.na(diff\_cash))  
  
# Test again after differencing  
atm2\_diff %>% features(diff\_cash, unitroot\_kpss)

## # A tibble: 1 × 3  
## ATM kpss\_stat kpss\_pvalue  
## <chr> <dbl> <dbl>  
## 1 ATM2 0.0104 0.1

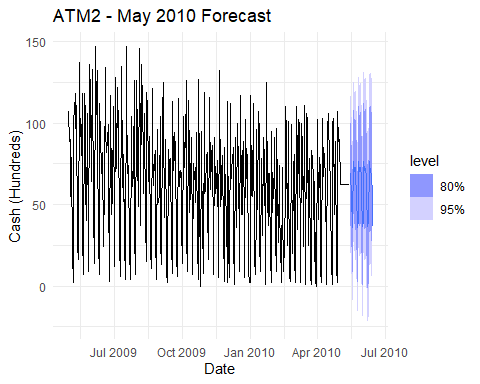
atm2\_model <- atm2\_data %>%  
 model(ARIMA = ARIMA(Cash))  
  
report(atm2\_model)

## Series: Cash   
## Model: ARIMA(2,0,2)(0,1,1)[7]   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 sma1  
## -0.4380 -0.9154 0.4921 0.8017 -0.7701  
## s.e. 0.0525 0.0383 0.0814 0.0560 0.0392  
##   
## sigma^2 estimated as 602.2: log likelihood=-1718.39  
## AIC=3448.78 AICc=3449.01 BIC=3472.29

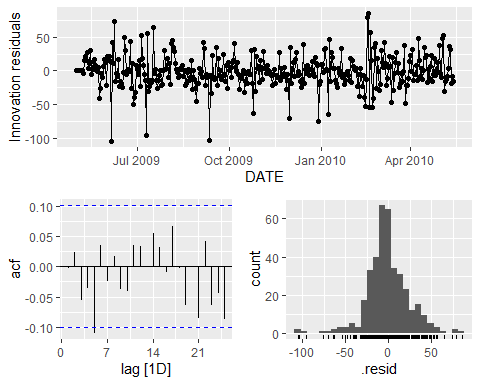
atm2\_model <- atm2\_data %>%  
 model(ARIMA = ARIMA(Cash))  
  
report(atm2\_model)

## Series: Cash   
## Model: ARIMA(2,0,2)(0,1,1)[7]   
##   
## Coefficients:  
## ar1 ar2 ma1 ma2 sma1  
## -0.4380 -0.9154 0.4921 0.8017 -0.7701  
## s.e. 0.0525 0.0383 0.0814 0.0560 0.0392  
##   
## sigma^2 estimated as 602.2: log likelihood=-1718.39  
## AIC=3448.78 AICc=3449.01 BIC=3472.29

atm2\_forecast <- atm2\_model %>%  
 forecast(h = "31 days")  
  
autoplot(atm2\_forecast, atm2\_data) +  
 labs(title = "ATM2 - May 2010 Forecast",  
 y = "Cash (Hundreds)", x = "Date") +  
 theme\_minimal()



atm2\_model %>%  
 gg\_tsresiduals()



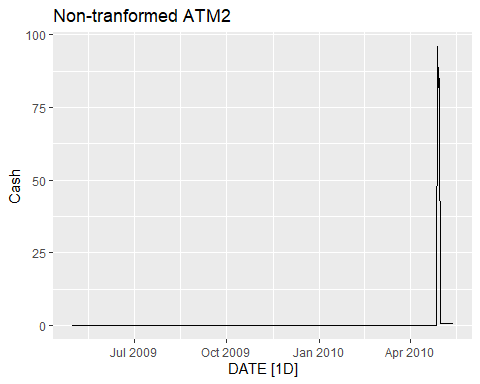
atm2\_model %>%  
 augment() %>%  
 features(.resid, ljung\_box, lag = 10)

## # A tibble: 1 × 4  
## ATM .model lb\_stat lb\_pvalue  
## <chr> <chr> <dbl> <dbl>  
## 1 ATM2 ARIMA 8.47 0.583

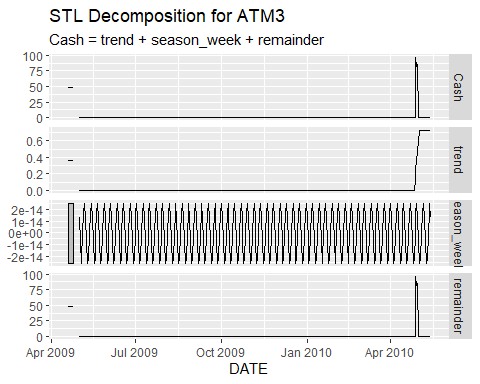
# print the files   
forecast\_atm2 <- atm2\_forecast %>%  
 as\_tibble() %>%  
 select(DATE, ATM, .mean) %>%  
 pivot\_wider(names\_from = ATM, values\_from = .mean)  
  
# Save to Excel  
write\_xlsx(forecast\_atm2, "ATM2\_Forecast\_May2010.xlsx")

### ATM3

ts\_data %>%  
 filter(ATM == "ATM3") %>%  
 autoplot(Cash) +  
 ggtitle("Non-tranformed ATM2")



ts\_data %>%  
 filter(ATM == "ATM3") %>%  
 model(STL(Cash ~ season(window = "periodic"), robust = TRUE)) %>%  
 components() %>%  
 autoplot() +  
 labs(title = "STL Decomposition for ATM3")



# --- 1. Extract ATM2 from data\_long ---  
atm3\_data <- ts\_data %>%  
 filter(ATM == "ATM3")  
  
# --- 2. Check if it's stationary (ADF test) ---  
atm3\_data %>% features(Cash, unitroot\_kpss)

## # A tibble: 1 × 3  
## ATM kpss\_stat kpss\_pvalue  
## <chr> <dbl> <dbl>  
## 1 ATM3 0.375 0.0880

Based on the grapgh, ATM3 appear to be new and lack historical pattern. The KPSS test was conducted on the ATM3 cash withdrawal time series to evaluate its stationarity. The test yielded a KPSS statistic of 0.375 with a corresponding p-value of 0.088. Since the p-value is greater than the conventional significance level of 0.05, we fail to reject the null hypothesis that the series is stationary. This indicates there is no strong evidence of a unit root, and the data can be considered stationary. Therefore, based on the KPSS test results, the ATM3 time series does not exhibit significant non-stationarity and may be suitable for modeling without further differencing.

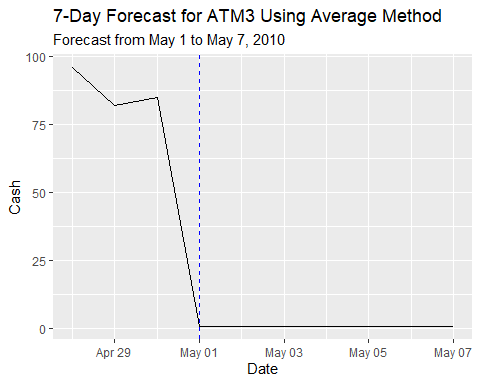
This ARIMA(0,0,2) model assumes the data is already stationary and captures the short-term autocorrelation using two moving average components. The model fits the data well, as indicated by the significant MA coefficients and relatively low AIC. It’s a suitable model for forecasting if the residual diagnostics confirm white noise behavior.

In Order to make this model better, we need more historical data. This prediction is only based on 3 days. Although an ARIMA(0,0,2) model was successfully fitted to the ATM3 cash withdrawal series, several factors indicate that the model should not be relied upon for meaningful forecasting. The historical data for ATM3 is extremely limited, with most values either zero or constant imputed values, and only a few days reflecting real activity. While the KPSS test suggests the series is stationary (p = 0.088), the residual diagnostics show a lack of meaningful variation, with nearly all residuals close to zero and a single large spike occurring near the end. This pattern suggests the model is likely overfitting a small number of recent values rather than capturing a true underlying pattern. The fitted MA coefficients may appear statistically significant, but in the context of such sparse and uninformative data, they are unlikely to generalize well. Therefore, despite the model’s statistical output, it is recommended to treat ATM3 as a newly active ATM and to defer reliable time series forecasting until more real data becomes available over time.

atm3\_real <- ts\_data %>%  
 filter(ATM == "ATM3", Cash != 0.7205479)  
head(atm3\_real)

## # A tsibble: 6 x 3 [1D]  
## # Key: ATM [1]  
## DATE ATM Cash  
## <date> <chr> <dbl>  
## 1 2009-05-01 ATM3 0  
## 2 2009-05-02 ATM3 0  
## 3 2009-05-03 ATM3 0  
## 4 2009-05-04 ATM3 0  
## 5 2009-05-05 ATM3 0  
## 6 2009-05-06 ATM3 0

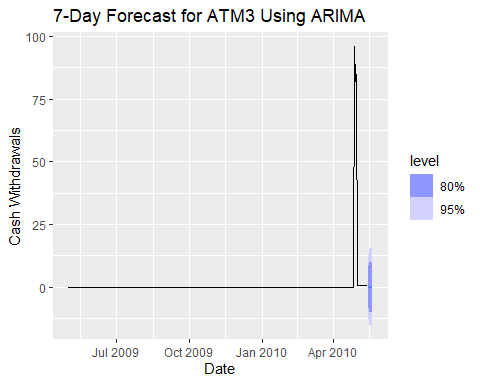
# Calculate average  
avg\_cash <- mean(atm3\_real$Cash)  
  
# Create future dates  
future\_dates <- seq(from = as.Date("2010-05-01"), by = "day", length.out = 30)  
  
# Create forecast dataframe  
atm3\_forecast <- tibble(  
 DATE = future\_dates,  
 ATM = "ATM3",  
 Cash = avg\_cash  
)  
  
# Step 4: Combine real and forecasted data  
combined <- bind\_rows(  
 as\_tibble(atm3\_real),  
 atm3\_forecast  
)  
  
# Filter to a 10-day window (3 actual + 7 forecast)  
plot\_data <- combined %>%  
 filter(DATE >= as.Date("2010-04-28") & DATE <= as.Date("2010-05-07"))  
  
# Plot  
ggplot(plot\_data, aes(x = DATE, y = Cash)) +  
 geom\_line(color = "black") +  
 geom\_vline(xintercept = as.Date("2010-05-01"), linetype = "dashed", color = "blue") +  
 labs(  
 title = "7-Day Forecast for ATM3 Using Average Method",  
 subtitle = "Forecast from May 1 to May 7, 2010",  
 y = "Cash", x = "Date"  
 )



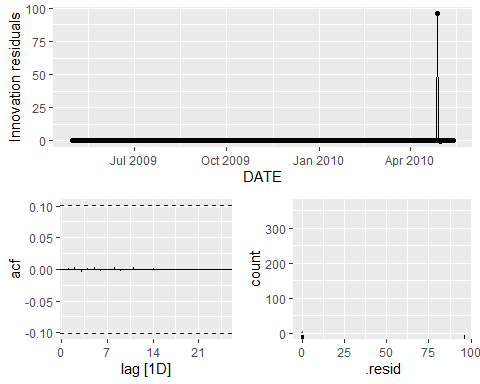
# Step 2: Fit ARIMA model (automatically selected orders)  
 atm3\_model <- atm3\_data %>%  
 model(ARIMA(Cash))  
   
 # Step 3: Model report  
 report(atm3\_model)

## Series: Cash   
## Model: ARIMA(0,0,2)   
##   
## Coefficients:  
## ma1 ma2  
## 0.8492 0.8752  
## s.e. 0.0263 0.0268  
##   
## sigma^2 estimated as 24.45: log likelihood=-1144.11  
## AIC=2294.23 AICc=2294.29 BIC=2306.04

# Step 4: Forecast 30 days ahead  
 forecast\_atm3 <- atm3\_model %>%  
 forecast(h = "7 days")  
   
 # Step 5: Plot forecast with historical data  
 autoplot(forecast\_atm3, atm3\_data) +  
 labs(title = "7-Day Forecast for ATM3 Using ARIMA",  
 y = "Cash Withdrawals", x = "Date")



# Step 6: Residual diagnostics  
 # a) ACF plot of residuals  
 atm3\_model %>%  
 gg\_tsresiduals()



# b) Ljung-Box test for white noise residuals  
 atm3\_model %>%  
 augment() %>%  
 features(.innov, ljung\_box, lag = 24, dof = 3)

## # A tibble: 1 × 4  
## ATM .model lb\_stat lb\_pvalue  
## <chr> <chr> <dbl> <dbl>  
## 1 ATM3 ARIMA(Cash) 0.0371 1

# extract point from forecast only   
 forecast\_atm3f <- forecast\_atm3 %>%  
 as\_tibble() %>%  
 select(DATE, ATM, .mean) %>%  
 pivot\_wider(names\_from = ATM, values\_from = .mean)  
   
 # Save to Excel  
 write\_xlsx(forecast\_atm3f, "ATM3\_Forecast\_May2010.xlsx")

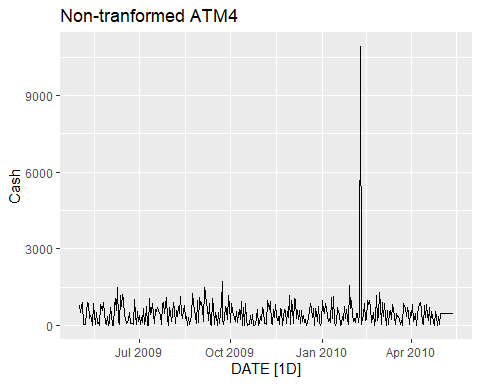
### ATM4

The time series plot of ATM4 displays daily cash withdrawals over the observed period, with most values remaining relatively consistent and moderate in scale. However, there is a clear and dramatic spike in withdrawals around early 2010, where values surge to nearly 10,000 (in hundreds of dollars), indicating a significant outlier or unusual event. Outside of this spike, the series exhibits mild variability with values largely below 1,000, suggesting stable usage with occasional fluctuations. This sharp peak skews the data and would likely impact forecasting models, highlighting the need to address or adjust outliers prior to model fitting.

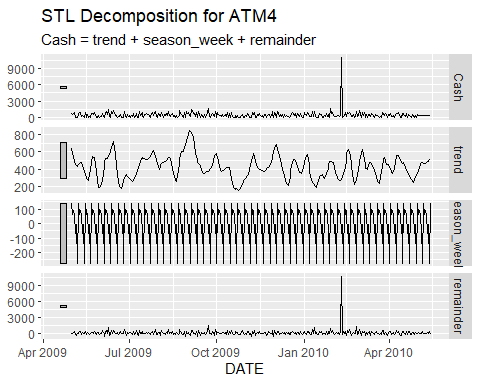
The null hypothesis of the KPSS test is that the time series is stationary. Because the p-value (0.018) is less than 0.05, we reject the null hypothesis.This indicates there ATM4 series is not stationary and requires differencing before fitting an ARIMA model and Applying first-order differencing (and possibly seasonal differencing) to make the series stationary before using ARIMA

After Dapply first order differencing, The the series is now stationary. Since p-value = 0.1 > 0.05, we fail to reject the null hypothesis. This confirms that the differenced series is now stationary, making it suitable for ARIMA modeling.

ts\_data %>%  
 filter(ATM == "ATM4") %>%  
 autoplot(Cash) +  
 ggtitle("Non-tranformed ATM4")



ts\_data %>%  
 filter(ATM == "ATM4") %>%  
 model(STL(Cash ~ season(window = "periodic"), robust = TRUE)) %>%  
 components() %>%  
 autoplot() +  
 labs(title = "STL Decomposition for ATM4")



atm4\_data <- ts\_data %>%  
 filter(ATM == "ATM4")  
head(atm4\_data)

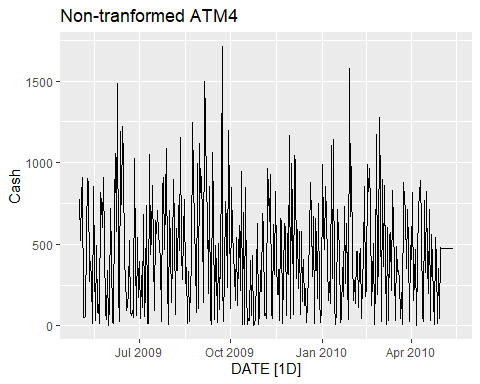
## # A tsibble: 6 x 3 [1D]  
## # Key: ATM [1]  
## DATE ATM Cash  
## <date> <chr> <dbl>  
## 1 2009-05-01 ATM4 777.   
## 2 2009-05-02 ATM4 524.   
## 3 2009-05-03 ATM4 793.   
## 4 2009-05-04 ATM4 908.   
## 5 2009-05-05 ATM4 52.8  
## 6 2009-05-06 ATM4 52.2

# --- 2. Check if it's stationary (ADF test) ---  
atm4\_data %>% features(Cash, unitroot\_kpss)

## # A tibble: 1 × 3  
## ATM kpss\_stat kpss\_pvalue  
## <chr> <dbl> <dbl>  
## 1 ATM4 0.0767 0.1

After replacing the maximum cash value in the ATM4 time series with the mean, the KPSS test was rerun to assess stationarity. The test returned a KPSS statistic of 0.0843 with a p-value of 0.1. Since the p-value exceeds the 0.05 threshold, we fail to reject the null hypothesis that the series is stationary. This indicates that the ATM4 cash withdrawal series remains stationary even after the outlier adjustment. The result confirms that the extreme value did not significantly affect the overall stationarity of the data. With the series now cleaner and statistically stable, it is suitable for further time series modeling and forecasting, such as using an ARIMA approach.

# Step 1: Filter ATM4 data  
atm4\_clean <- atm4\_data %>%  
 group\_by(ATM) %>%  
 mutate(  
 # Step 2: Calculate the mean (excluding NA and extreme value)  
 mean\_cash = mean(Cash[ATM == "ATM4"], na.rm = TRUE),  
   
 # Step 3: Identify max value  
 max\_cash = max(Cash[ATM == "ATM4"], na.rm = TRUE),  
   
 # Step 4: Replace max with mean  
 Cash = ifelse(ATM == "ATM4" & Cash == max\_cash, mean\_cash, Cash)  
 ) %>%  
 select(-mean\_cash, -max\_cash) # Optional: remove helper columns  
# Convert to tsibble  
  
atm4\_clean <- atm4\_clean %>%  
 as\_tsibble(index = DATE)  
  
  
  
  
  
  
atm4\_clean %>%  
 autoplot(Cash) +  
 ggtitle("Non-tranformed ATM4")



# --- 2. Check if it's stationary (ADF test) ---  
atm4\_clean %>% features(Cash, unitroot\_kpss)# --- 2. Check if it's stationary (ADF test) ---

## # A tibble: 1 × 3  
## ATM kpss\_stat kpss\_pvalue  
## <chr> <dbl> <dbl>  
## 1 ATM4 0.100 0.1

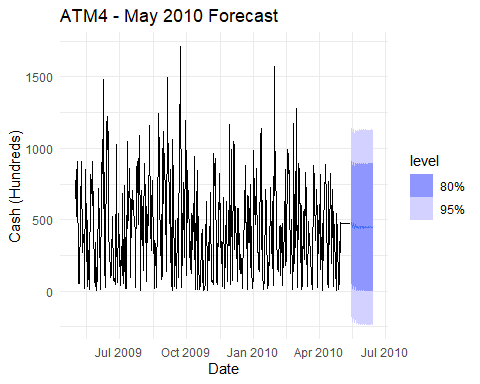
An ARIMA(3,0,2)(1,0,0)[7] model was fitted to the cleaned ATM4 cash withdrawal series, accounting for both non-seasonal and seasonal components with a weekly period. The model includes three autoregressive terms (AR1–AR3), two moving average terms (MA1–MA2), and one seasonal AR term, all of which are statistically significant with relatively low standard errors. The residual variance was estimated at 108,412 with a log-likelihood of -2731.51, and model selection criteria were AIC = 5479.02 and BIC = 5510.52. The 30-day forecast for May 2010 reflects stable cash withdrawal patterns with reasonable confidence intervals. Residual diagnostics indicate that the residuals are roughly centered around zero with no major autocorrelation, as shown by the flat ACF plot. Furthermore, the Ljung-Box test returned a p-value of 0.393, indicating no significant autocorrelation remaining in the residuals. Overall, the model appears to provide a good fit to the data and is appropriate for short-term forecasting of ATM4 cash withdrawals.

atm4\_model <- atm4\_clean %>%  
 model(ARIMA = ARIMA(Cash))  
  
report(atm4\_model)

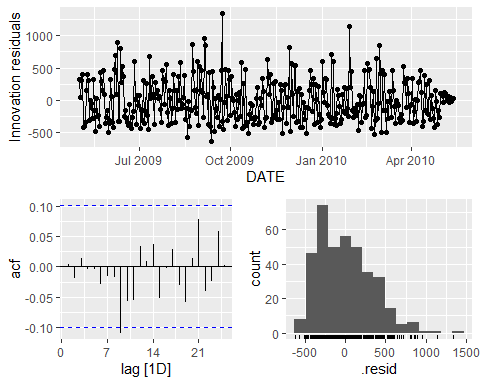
## Series: Cash   
## Model: ARIMA(3,0,2)(1,0,0)[7] w/ mean   
##   
## Coefficients:  
## ar1 ar2 ar3 ma1 ma2 sar1 constant  
## -0.8855 -0.6709 0.1652 0.9692 0.7935 0.2446 806.7806  
## s.e. 0.1045 0.1238 0.0553 0.0957 0.1155 0.0542 47.0171  
##   
## sigma^2 estimated as 113290: log likelihood=-2739.82  
## AIC=5495.64 AICc=5496.03 BIC=5527.14

atm4\_forecast <- atm4\_model %>%  
 forecast(h = "31 days")  
  
autoplot(atm4\_forecast, atm4\_clean) +  
 labs(title = "ATM4 - May 2010 Forecast",  
 y = "Cash (Hundreds)", x = "Date") +  
 theme\_minimal()

## `mutate\_if()` ignored the following grouping variables:  
## • Column `ATM`



atm4\_model %>% gg\_tsresiduals()



# b) Ljung-Box test for white noise residuals  
 atm4\_model %>%  
 augment() %>%  
 features(.innov, ljung\_box, lag = 24, dof = 3)

## # A tibble: 1 × 4  
## ATM .model lb\_stat lb\_pvalue  
## <chr> <chr> <dbl> <dbl>  
## 1 ATM4 ARIMA 16.9 0.718

# extract point from forecast only   
 forecast\_atm4 <- atm4\_forecast %>%  
 as\_tibble() %>%  
 select(DATE, Cash, .mean) %>%  
 pivot\_wider(names\_from = Cash, values\_from = .mean)  
  
   
 # Save to Excel  
 write\_xlsx(forecast\_atm3f, "ATM4\_Forecast\_May2010.xlsx")

## **Part B - Forecasting Power**

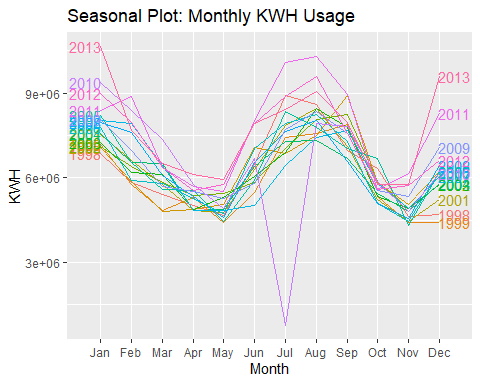
Part B consists of a simple dataset of residential power usage for January 1998 until December 2013. Your assignment is to model these data and a monthly forecast for 2014. The data is given in a single file. The variable ‘KWH’ is power consumption in Kilowatt hours, the rest is straight forward

The seasonal plot of monthly residential electricity consumption from 1998 to 2013 reveals a clear and recurring seasonal pattern. Power usage tends to peak during the winter and summer months, particularly in **January** and **July–August**, and drops during the spring and fall. This seasonal behavior is largely influenced by **temperature-dependent energy demands**, such as heating in the colder months and air conditioning during warmer months. Given this consistent yearly cycle, it is essential to apply forecasting models that can accurately capture both **seasonal fluctuations** and **long-term trends**.

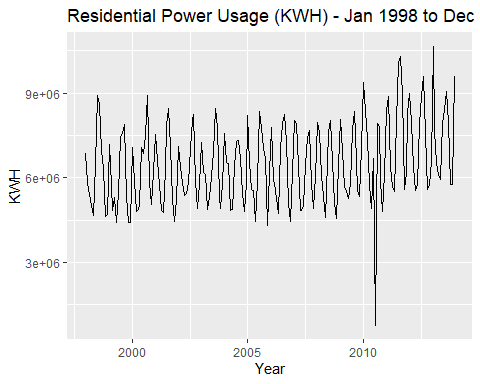
To address this, we employ two widely used time series forecasting methods: **ETS (Error, Trend, Seasonal)** and **Auto ARIMA**. The **ETS model** is particularly effective for data with strong trend and seasonality components, automatically selecting the best configuration of exponential smoothing. Meanwhile, **Auto ARIMA** (Auto-Regressive Integrated Moving Average) automatically identifies the optimal seasonal and non-seasonal parameters, making it well-suited for data with autocorrelation and seasonality. By applying both models, we can compare their forecasting accuracy and choose the one that best fits the underlying structure of the data.

.

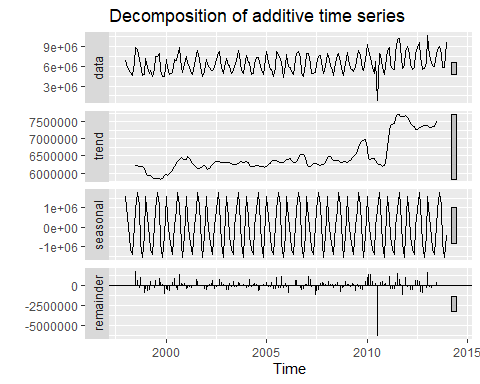
# Load the Excel file  
data <- read\_excel("ResidentialCustomerForecastLoad-624.xlsx", sheet = "ResidentialCustomerForecastLoad")  
  
# Convert the 'YYYY-MMM' column to Date  
data <- data %>%  
 mutate(Date = parse\_date\_time(`YYYY-MMM`, orders = "ym")) %>%  
 arrange(Date)  
  
# Replace missing KWH value using linear interpolation  
data$KWH <- zoo::na.approx(data$KWH)  
  
  
# Convert to time series object (monthly frequency starting from Jan 1998)  
kwh\_ts <- ts(data$KWH, start = c(1998, 1), frequency = 12)  
  
  
  
  
# Check for seasonality  
  
  
# Plot seasonal subseries  
ggseasonplot(kwh\_ts, year.labels = TRUE, year.labels.left = TRUE) +  
 ggtitle("Seasonal Plot: Monthly KWH Usage") +  
 ylab("KWH")



# Plot the time series  
autoplot(kwh\_ts) +  
 ggtitle("Residential Power Usage (KWH) - Jan 1998 to Dec 2013") +  
 xlab("Year") + ylab("KWH")



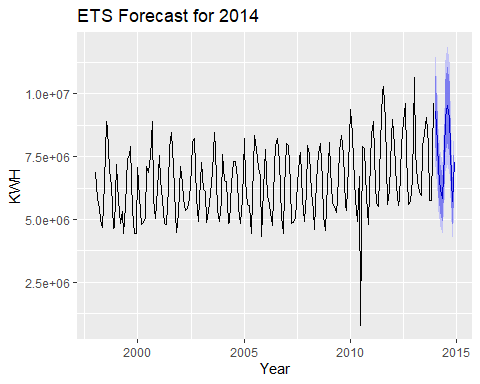
# Decompose the time series  
decomposed <- decompose(kwh\_ts)  
autoplot(decomposed)



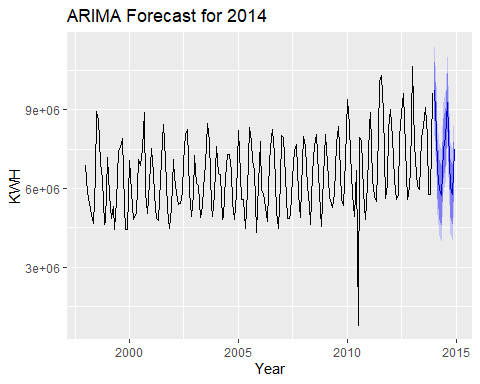
The accuracy metrics from the four models—two standard and two with Box-Cox transformation—provide insight into the effect of variance stabilization on forecast performance for residential power usage. The original ARIMA model (without transformation) performed the best overall, with the lowest RMSE (827,254.2), MAE (493,308.5), and MAPE (11.69%). It also showed minimal autocorrelation in residuals (ACF1 = 0.0128), indicating a well-fitted model. The ETS model (without transformation) followed closely with slightly higher RMSE (835,107) and MAPE (12.04%), but still maintained reasonable accuracy and fit.

However, the models using Box-Cox transformation (ETS model 2 and ARIMA model 2) did not improve forecast accuracy. In fact, **ARIMA model 2**, which applied the Box-Cox transformation, performed the worst with significantly higher RMSE (1,197,097), MAE (855,261.8), and MAPE (16.38%). Similarly, **ETS model 2** showed increased error compared to its untransformed version, with an RMSE of 876,478.9 and MAPE of 11.92%. These results suggest that applying the Box-Cox transformation in this case may have distorted the relationship in the data or introduced unnecessary complexity. Therefore, the untransformed **ARIMA model** remains the most accurate and reliable choice for forecasting residential power consumption in this dataset.

# Forecast using ETS (Exponential Smoothing)  
ets\_model <- ets(kwh\_ts)  
ets\_forecast <- forecast(ets\_model, h = 12)  
autoplot(ets\_forecast) +  
 ggtitle("ETS Forecast for 2014") +  
 xlab("Year") + ylab("KWH")



# Forecast using auto.arima (optional)  
arima\_model <- auto.arima(kwh\_ts)  
arima\_forecast <- forecast(arima\_model, h = 12)  
autoplot(arima\_forecast) +  
 ggtitle("ARIMA Forecast for 2014") +  
 xlab("Year") + ylab("KWH")



accuracy(ets\_model)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 61009.73 835107 503972.9 -4.39013 12.04006 0.7233624 0.1698584

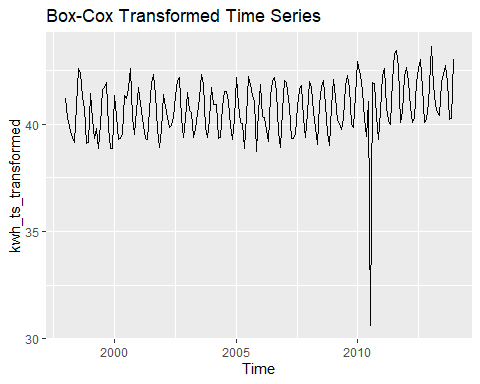
accuracy(arima\_model)

## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set -25089.69 827254.2 493308.5 -5.511184 11.685 0.7080556 0.01283694

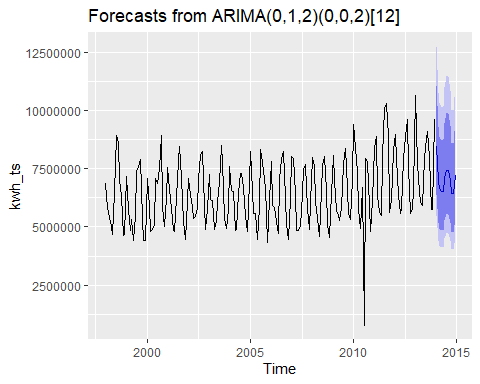
lambda <- BoxCox.lambda(kwh\_ts)  
lambda

## [1] 0.1074118

kwh\_ts\_transformed <- BoxCox(kwh\_ts, lambda)  
autoplot(kwh\_ts\_transformed) +  
 ggtitle("Box-Cox Transformed Time Series")



ets\_model2 <- ets(kwh\_ts, lambda = lambda)  
  
arima\_model2 <- auto.arima(kwh\_ts, lambda = lambda)  
  
forecast\_arima2 <- forecast(arima\_model2, h = 12)  
autoplot(forecast\_arima2)



# Create a data frame of the forecasts  
forecast\_df <- data.frame(  
 Month = seq(as.Date("2014-01-01"), by = "month", length.out = 12),  
 ETS = ets\_forecast$mean,  
 ARIMA = arima\_forecast$mean  
)  
  
# Write to CSV  
write.csv(forecast\_df, "PartB\_Forecast\_2014.csv", row.names = FALSE)

## **Part C – BONUS**

Part C consists of two data sets. These are simple 2 columns sets, however they have different time stamps. Your optional assignment is to time-base sequence the data and aggregate based on hour (example of what this looks like, follows). Note for multiple recordings within an hour, take the mean. Then to determine if the data is stationary and can it be forecast. If so, provide a week forward forecast and present results via Rpubs and .rmd and the forecast in an Excel readable file.

We read the hourly water flow data from two Excel files (Waterflow\_Pipe1.xlsx and Waterflow\_Pipe2.xlsx), processes them to compute the average hourly flow per day, and merges them into a single dataset. It ensures the timestamps are properly formatted and handles any missing values by replacing them with zero. Finally, it combines the two pipes’ flow values into a total water flow (Water) at each hourly timestamp.

# Load the datasets  
pipe1 <- read\_excel("Waterflow\_Pipe1.xlsx", col\_types = c("date", "numeric"))  
pipe2 <- read\_excel("Waterflow\_Pipe2.xlsx", col\_types = c("date", "numeric"))  
  
colnames(pipe1) <- c("DateTime", "WaterFlow")   
colnames(pipe2) <- c("DateTime", "WaterFlow")   
  
  
  
pipe1 <- pipe1 %>% mutate(Date = as.Date(DateTime), Time = hour(DateTime)+1) %>%   
 group\_by(Date, Time) %>%  
 summarise(Water=mean(WaterFlow)) %>%  
 ungroup() %>%  
 mutate(DateTime=ymd\_h(paste(Date,Time))) %>%  
 select(DateTime,Water)

## `summarise()` has grouped output by 'Date'. You can override using the  
## `.groups` argument.

pipe2 <- pipe2 %>% mutate(Date = as.Date(DateTime), Time = hour(DateTime)+1) %>%   
 group\_by(Date, Time) %>%  
 summarise(Water=mean(WaterFlow)) %>%  
 ungroup() %>%  
 mutate(DateTime=ymd\_h(paste(Date,Time))) %>%  
 select(DateTime,Water)

## `summarise()` has grouped output by 'Date'. You can override using the  
## `.groups` argument.

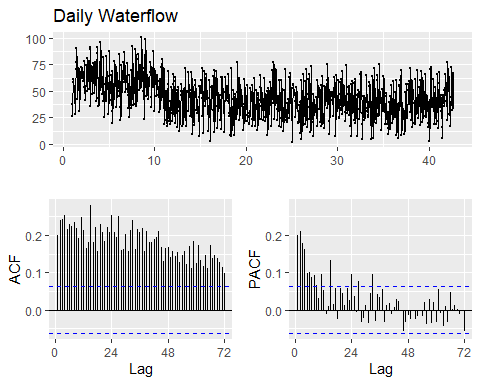
pipe2

## # A tibble: 1,000 × 2  
## DateTime Water  
## <dttm> <dbl>  
## 1 2015-10-23 02:00:00 18.8   
## 2 2015-10-23 03:00:00 43.1   
## 3 2015-10-23 04:00:00 38.0   
## 4 2015-10-23 05:00:00 36.1   
## 5 2015-10-23 06:00:00 31.9   
## 6 2015-10-23 07:00:00 28.2   
## 7 2015-10-23 08:00:00 9.86  
## 8 2015-10-23 09:00:00 26.7   
## 9 2015-10-23 10:00:00 55.8   
## 10 2015-10-23 11:00:00 54.2   
## # ℹ 990 more rows

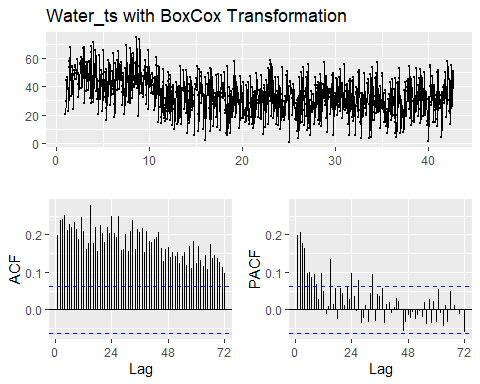
# Convert timestamp column to datetime format  
pipe1$DateTime <- ymd\_hms(pipe1[[1]])  
pipe2$DateTime <- ymd\_hms(pipe2[[1]])  
  
  
  
  
water <- full\_join(pipe1, pipe2, by="DateTime", suffix=c("\_1", "\_2")) %>%  
 mutate(Water\_1=ifelse(is.na(Water\_1), 0, Water\_1)) %>%  
 mutate(Water\_2=ifelse(is.na(Water\_2), 0, Water\_2)) %>%  
 mutate(Water = Water\_1 + Water\_2) %>%  
 select(DateTime, Water)

The stationarity of the combined hourly water flow series was evaluated using the KPSS (Kwiatkowski–Phillips–Schmidt–Shin) test. The test statistic was 0.0097, which is well below all conventional critical values (e.g., 0.347 at the 10% level). As the KPSS test has a null hypothesis of stationarity, we fail to reject the null, indicating that the series is stationary. This result supports the use of ARIMA modeling for forecasting, as stationarity is a key assumption for reliable time series predictions.

# Check for gaps  
water\_ts <- ts(water$Water, frequency=24)  
ggtsdisplay(water\_ts, main="Daily Waterflow")



water\_bc <- BoxCox(water\_ts, lambda = BoxCox.lambda(water\_ts))  
ggtsdisplay(water\_bc, main="Water\_ts with BoxCox Transformation")



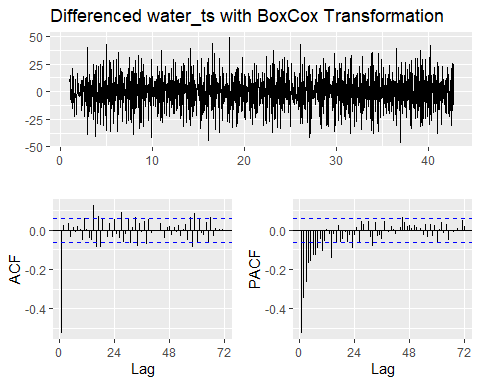
ndiffs(water\_ts)

## [1] 1

nsdiffs(water\_ts)

## [1] 0

ggtsdisplay(diff(water\_bc), points=FALSE, main="Differenced water\_ts with BoxCox Transformation")



water\_bc %>% diff() %>% ur.kpss() %>% summary()

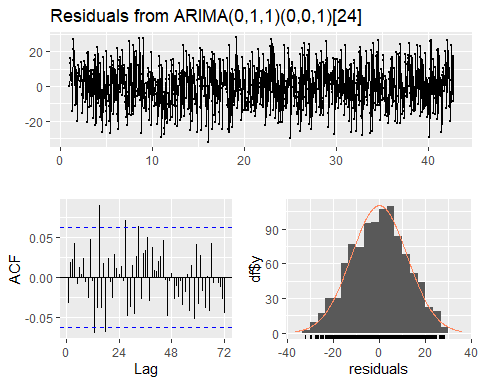
##   
## #######################   
## # KPSS Unit Root Test #   
## #######################   
##   
## Test is of type: mu with 7 lags.   
##   
## Value of test-statistic is: 0.0097   
##   
## Critical value for a significance level of:   
## 10pct 5pct 2.5pct 1pct  
## critical values 0.347 0.463 0.574 0.739

An ARIMA(0,1,1)(0,0,1)[24] model with a Box-Cox transformation (λ = 0.9199) was fitted to the hourly water flow series. The model includes a non-seasonal moving average term (MA(1)) and a seasonal moving average term (SMA(1)) with a 24-hour seasonal period. Both coefficients are statistically significant with low standard errors. The model produced a log-likelihood of -3913.24 and an AIC of 7832.48, indicating a good fit relative to complexity. Forecast accuracy metrics show a root mean squared error (RMSE) of 16.26 and a mean absolute percentage error (MAPE) of 50.23%. The Ljung-Box test on the residuals yielded a p-value of 0.05988, suggesting no significant autocorrelation remains in the residuals at the 5% level. Overall, the model appears adequate for short-term forecasting of hourly water flow.

water\_arima <- Arima(water\_ts, order=c(0,1,1), seasonal=c(0,0,1), lambda = BoxCox.lambda(water\_ts))  
summary(water\_arima)

## Series: water\_ts   
## ARIMA(0,1,1)(0,0,1)[24]   
## Box Cox transformation: lambda= 0.9199428   
##   
## Coefficients:  
## ma1 sma1  
## -0.9568 0.0659  
## s.e. 0.0104 0.0320  
##   
## sigma^2 = 146.7: log likelihood = -3913.24  
## AIC=7832.48 AICc=7832.5 BIC=7847.2  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.131665 16.2623 13.23158 -28.37285 50.22864 0.7447631 -0.03344016

checkresiduals(water\_arima)

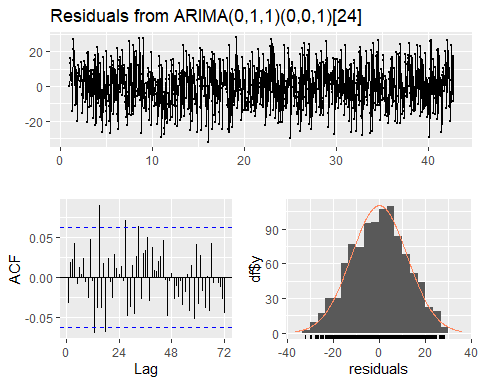


##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1)(0,0,1)[24]  
## Q\* = 61.785, df = 46, p-value = 0.05988  
##   
## Model df: 2. Total lags used: 48

water\_auto <- auto.arima(water\_ts, approximation = FALSE, lambda=BoxCox.lambda(water\_ts))  
summary(water\_auto)

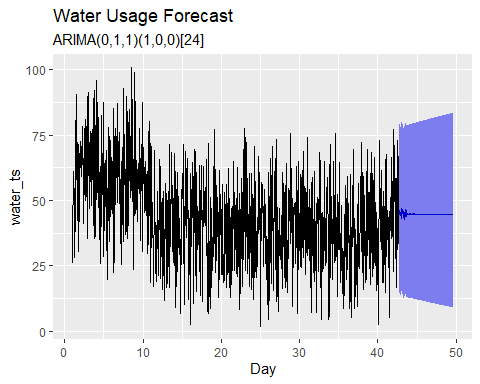
## Series: water\_ts   
## ARIMA(0,1,1)(0,0,1)[24]   
## Box Cox transformation: lambda= 0.9199428   
##   
## Coefficients:  
## ma1 sma1  
## -0.9568 0.0659  
## s.e. 0.0104 0.0320  
##   
## sigma^2 = 146.7: log likelihood = -3913.24  
## AIC=7832.48 AICc=7832.5 BIC=7847.2  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 0.131665 16.2623 13.23158 -28.37285 50.22864 0.7447631 -0.03344016

checkresiduals(water\_auto)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,1)(0,0,1)[24]  
## Q\* = 61.785, df = 46, p-value = 0.05988  
##   
## Model df: 2. Total lags used: 48

water\_model <- Arima(water\_ts, order=c(0,1,1), seasonal=c(1,0,0), lambda = BoxCox.lambda(water\_ts))  
water\_forecast <- forecast(water\_model, h=7\*24, level=95)  
autoplot(water\_forecast) +  
 labs(title="Water Usage Forecast", subtitle = "ARIMA(0,1,1)(1,0,0)[24]", x="Day")



library(tibble)  
  
  
# Convert to data frame  
forecast\_df <- as.data.frame(water\_forecast)  
  
forecast\_export <- forecast\_df %>%  
 rownames\_to\_column(var = "DateTime") %>%  
 rename(  
 Forecast = `Point Forecast`,  
 Lower\_95 = `Lo 95`,  
 Upper\_95 = `Hi 95`  
 )  
  
write\_xlsx(forecast\_export, "WaterFlow\_ARIMA\_Forecast.xlsx")

The ARIMA(0,1,1)(1,0,0)[24] model with a Box-Cox transformation was used to forecast hourly water flow over a 7-day horizon. The forecast plot shows the expected values with a 95% confidence interval that appropriately widens over time, reflecting increasing uncertainty. Residual diagnostics indicate that the residuals are centered around zero, display no clear patterns over time, and exhibit approximate normality as shown by the histogram and density overlay. The autocorrelation function (ACF) plot of the residuals suggests minimal serial correlation, with most lags falling within the 95% confidence bounds. These diagnostics collectively support that the model residuals behave like white noise, indicating a good model fit and reliable forecasting performance.