DATA624_Project1

John Ferrara

2025-03-23

Part A - ATM FORECAST (ATM624Data.xlsx)

Description

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 to make this have a little more business feeling. Explain and demonstrate your process, techniques used and not used, and your actual forecast. I am giving you data via an excel file, please provide your written report on your findings, visuals, discussion and your R code via an RPubs link along with the actual.rmd file Also please submit the forecast which you will put in an Excel readable file.

Overview

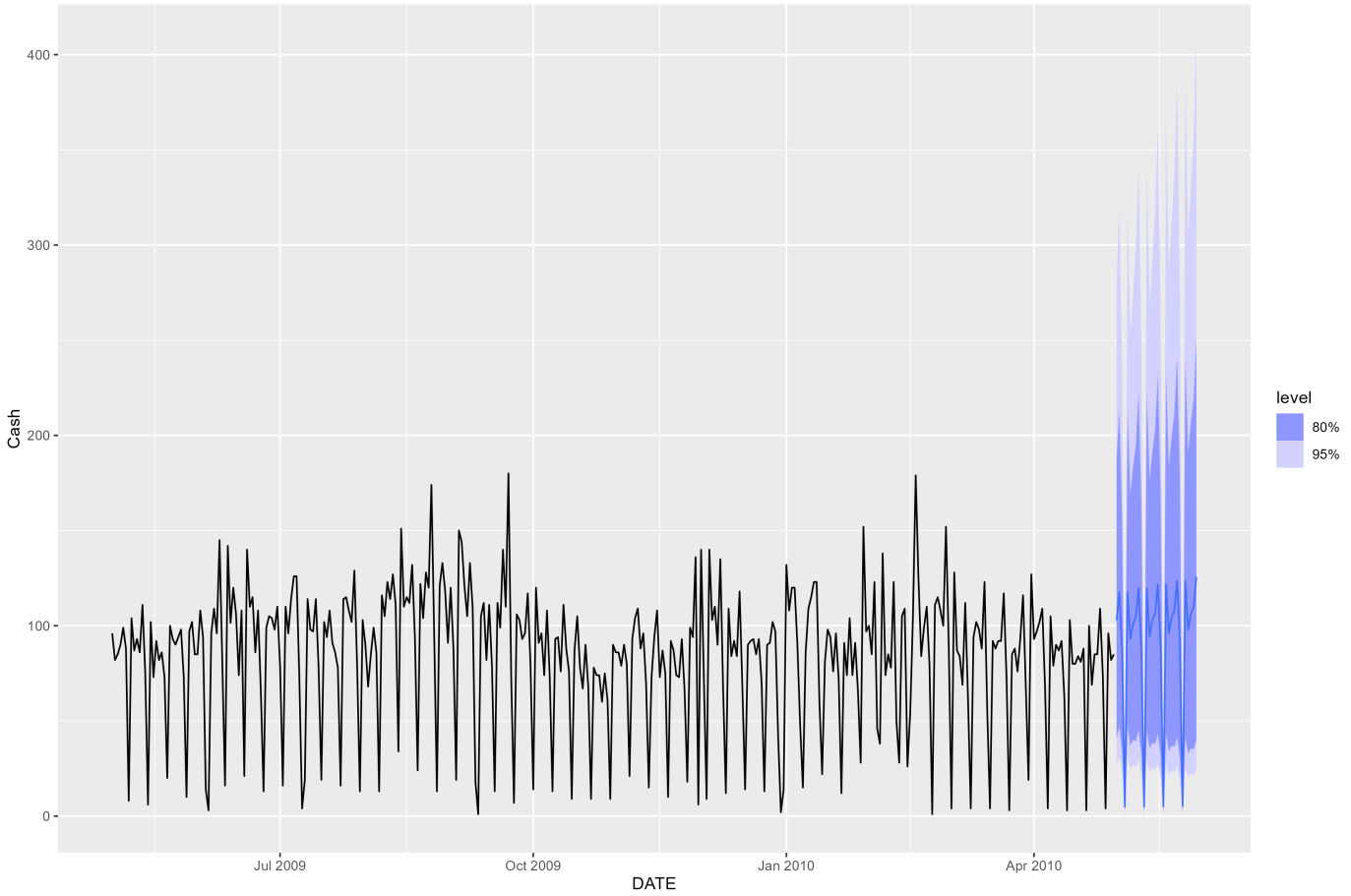
There were four different ATMs in the raw dataset. In order to accurately provide forecasts for each individual ATM machine and its respective cash flow the data was parsed so as to separate each of the four ATMs into their own individual datasets. However, before this data was parsed out based on the ATM number several steps were taken to prepare the data for analysis. The dataset was checked for null values. A total of 19 rows out of 1,474 total rows were found to have null values. Of these 19, there were 14 rows that had a null value for the ATM and the Cash value. These rows lacked enough data to be deemed useful towards the analysis and were dropped. The four remaining rows that contained null values were limited to the Cash column. Due to these rows having an ATM value they were left in. The 5 remaining null values impacted ATM1 and ATM2. In order to fill in these values, the average of the Cash values for the following and preceeding days were averaged in order to imputes these values. For instance, one of the null values found that impacted ATM 1 was a null cash value on 6/13/2009. The average of the cash values for ATM1 on 6/12/2009 and 6/14/2009 were averaged to impute the null value on 6/13/2009. This technique was used on each of the 5 total null cash values.

In addition to imputing null values in the data, the dates, having been sourced from and Excel file, needed formatting. In order to convert the dates from the raw excel format research into Excel’s data methodology had to be completed in order to identify the origin date that the program uses to count days. It was discovered that while Excel uses 1900/01/01 as base line date, the program also incorrectly considers 1900 a leap year. This mandates the use of a different origin date for proper date conversions. The date used was ‘1899-12-30’. Using this date the numbers initially read in were successfully converted to dates to use in the forecast analysis.

These were the main processing methodologies used on the data. The following sections address the methodology for May 2010 cash withdrawal projections for each of the ATMs.

ATM1

The following image is the initial data concerning ATM1 when plotted. As you can see there is no trend but plenty of seasonality on the weekly level.



Raw ATM1 Plot

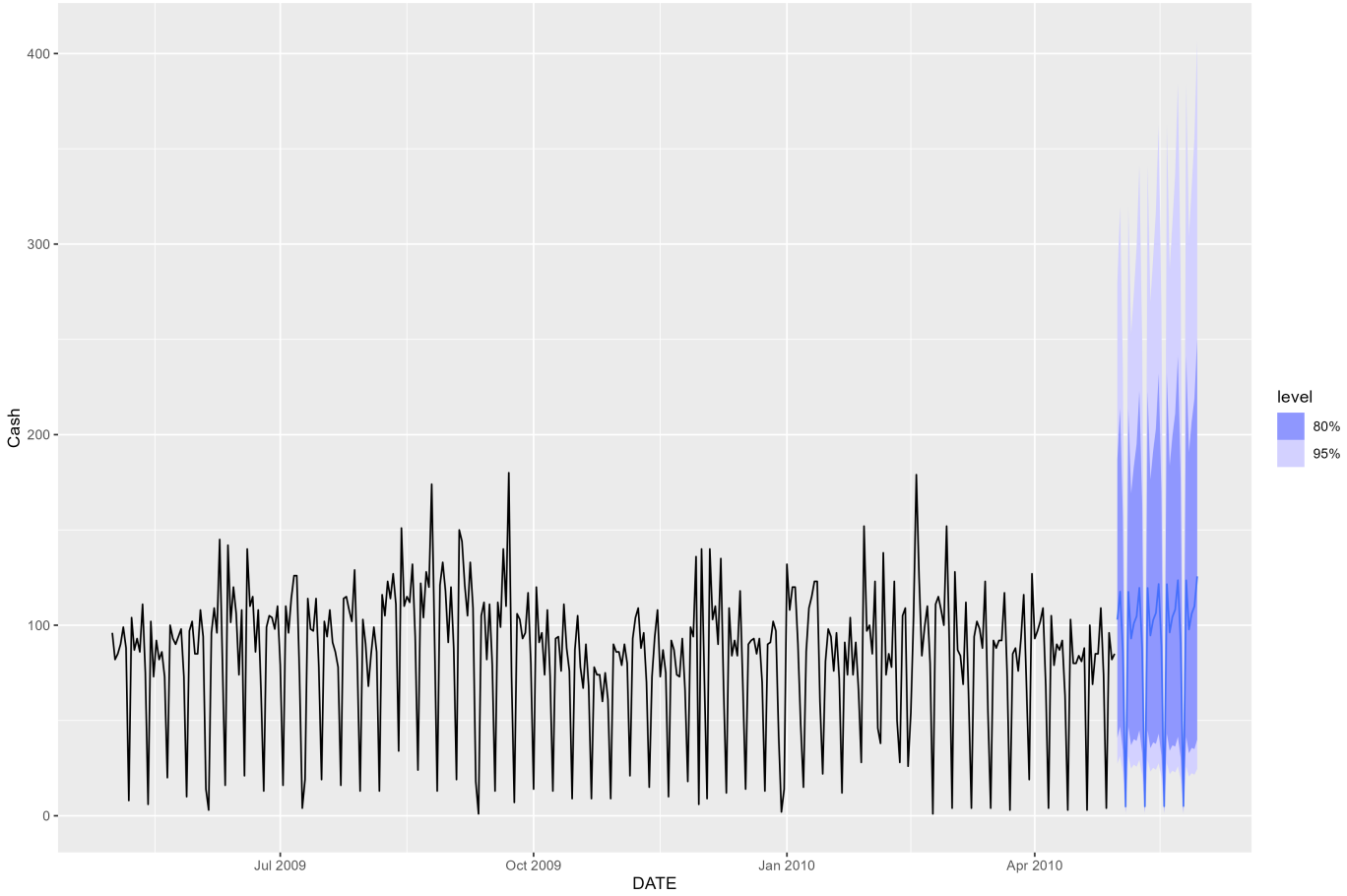
After checking for the presence of zeros in the Cash column, as this would impact certain transformation types. I decided to log the data in order to yeild a better performing model. I then proceeded to attempt different modeling techniques to yield an accurate projection. I tried to main modeling methodologies: ETS and ARIMA modeling. There were no zeros in the ATM1 Cash data column, so I manually attempted both additive and multiplicative seasonalities, while also using the function's native ability to identify the best recommended model. The following table breaks down the ETS and ARIMA models I attempted, along with the model I ultimately selected based on performance.

ATM 1 Model Comparison Table (ETS and ARIMA)

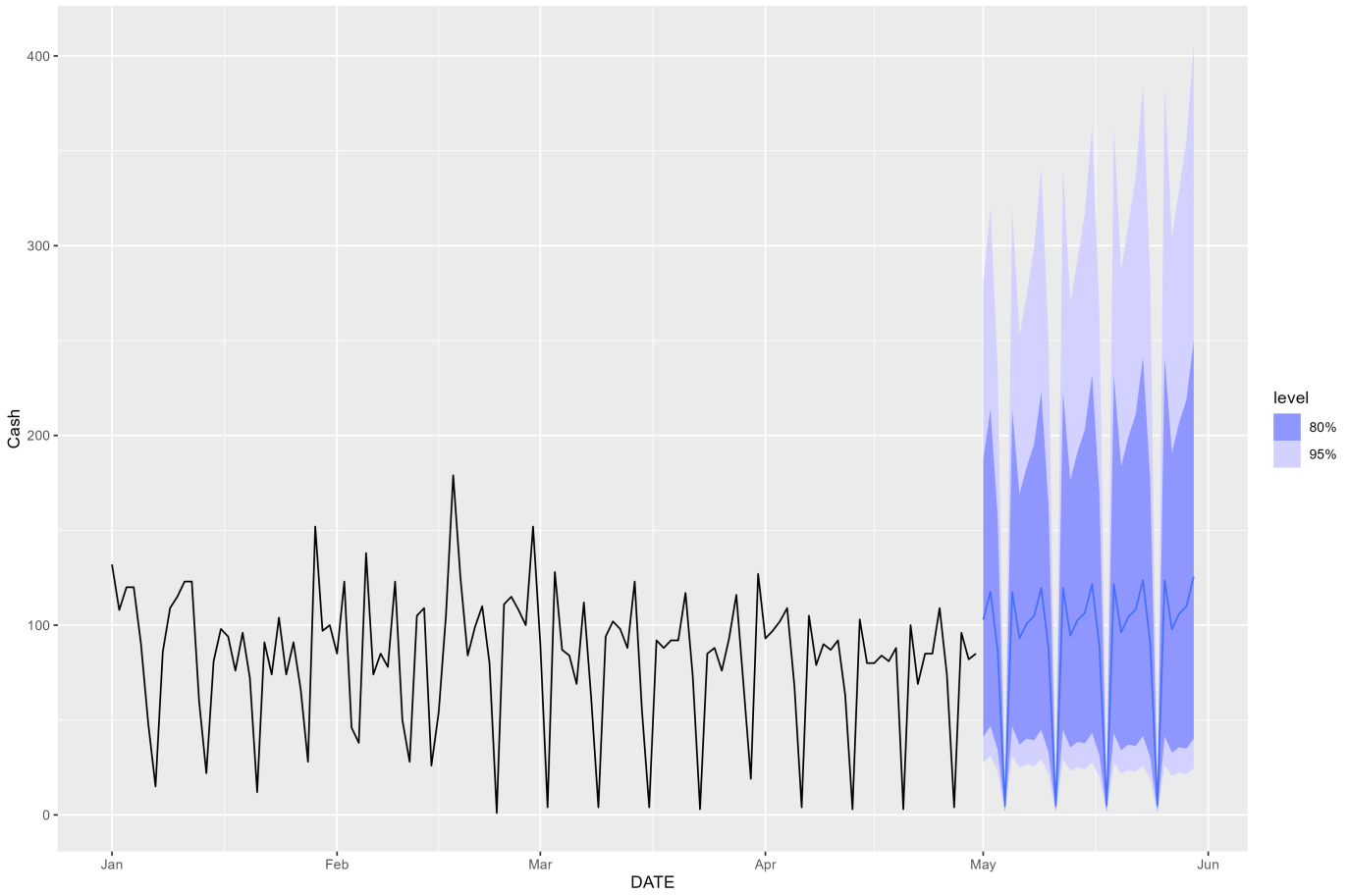
Model	Model Type	AIC	AICc	BIC	RMSE	Status
auto_ANA ETA(ANA)	ETS	4488.413	4489.035	4527.412	23.83524	Rejected
ANM	ETS	4488.277	4488.898	4527.276	23.83080	Rejected
MNM	ETS	4566.787	4567.408	4605.786	26.34075	Rejected
MNA	ETS	4595.487	4596.108	4634.486	23.99628	Rejected
manual_select2 ARIMA(0,0,0)(2,1,0)	ARIMA	670.4853	670.5531	682.1269	27.30840	Rejected
manual_select3 ARIMA(0,0,0)(3,1,0)	ARIMA	661.1413	661.2547	676.6635	26.75715	Rejected

Model	Model Type	AIC	AICc	BIC	RMSE	Status
auto_step ARIMA(0,0,0)(0,1,1)	ARIMA	647.8779	647.9117	655.6390	25.99424	Selected
auto_search ARIMA(0,0,2)(0,1,1)	ARIMA	646.3709	646.4842	661.8930	26.42130	Rejected

Using the selected ARIMA model (ARIMA(0,0,0)(0,1,1)), a forecast for ATM1's cash output was made for 30 days after the end of the data, essentially May 2010. This was done after last confirmation of the model's residuals to not have any autocorrelations or odd patterns present by visually looking at the residuals and performing a LJung Box test to reteive a p-value. The projection image can be seen below, with the detailed daily projected values in the accompanying file in the ATM1 tab.



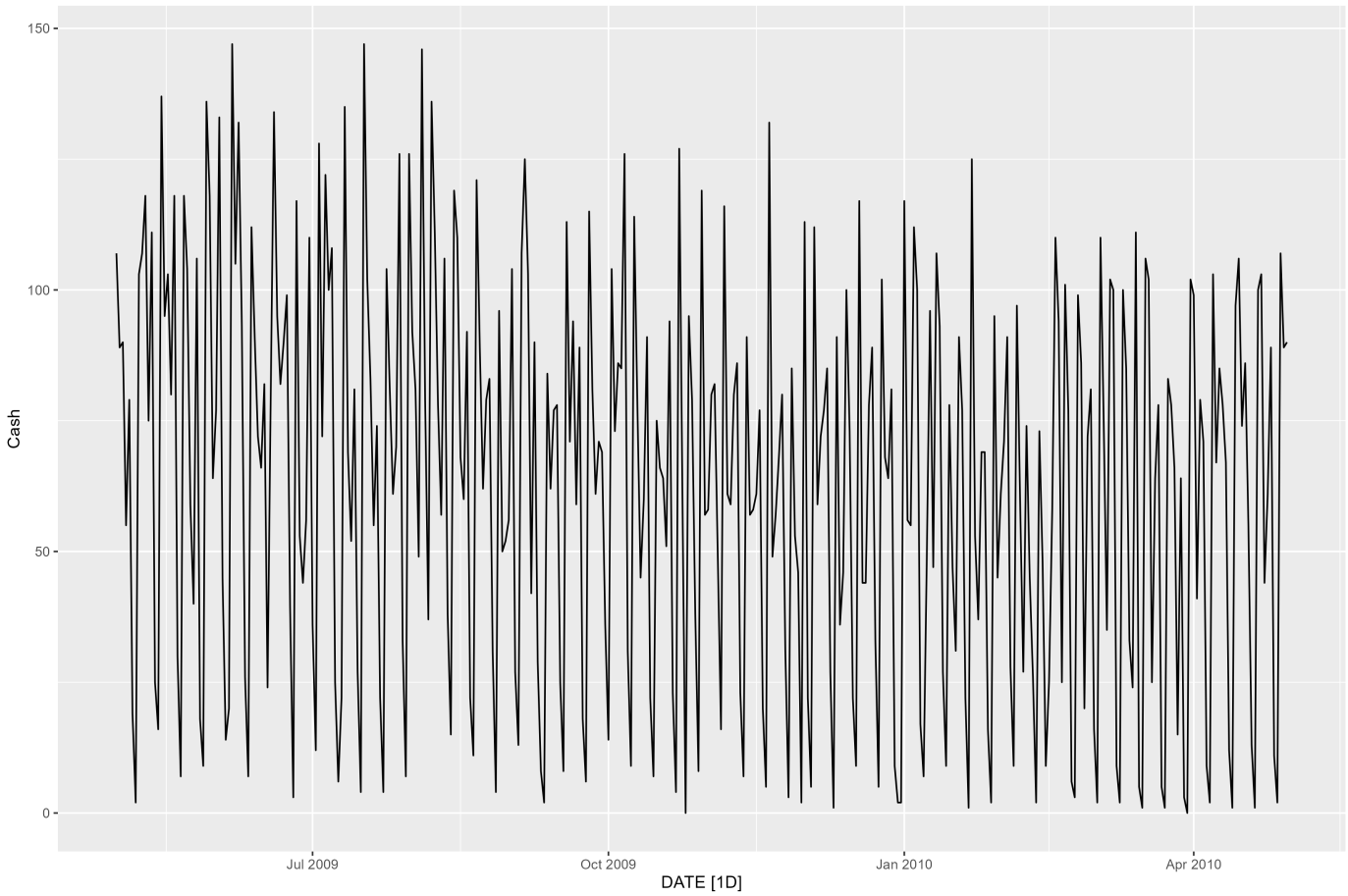
Full ATM1 Projection Forecast Plot



Overall, the projection predicts that for the month of may the cash withdrawals will be around $10,000 (as keep in mind the cash column is in hundreds). This does fluctuate a bit, but on 5/1/2010 the predicted value is about 103 for the Cash column, by the 15th of the month the prediction is around 106, and lastly by months end the prediction is ~125. More details can be found in the accompanying Excel file in the PARTA_ATM1 tab. Lastly, all of my code and work for this ATM can be foudn in Appendix A.

ATM2

The following image is the initial data concerning ATM2 when plotted. As you can see, similar to ATM1 there is no trend but plenty of seasonality on the weekly level.



Raw ATM2 Plot

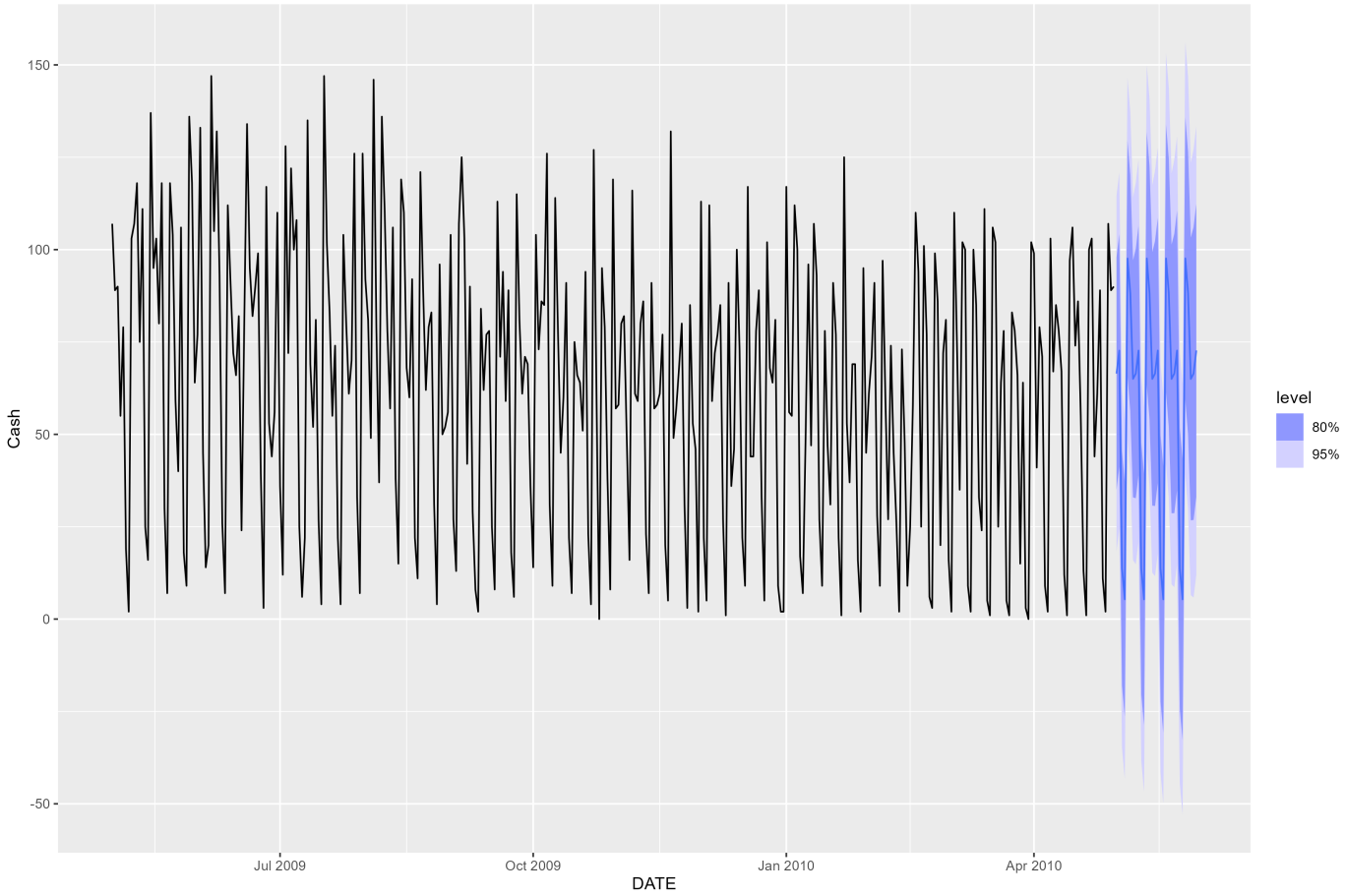
For this ATM there were 2 zero values in the cash column, which meant without transformation muiltiplicative ETS modeling was not possible. I performed the same process as I did with ATM 1, that is generating several ETS and ARIMA models in order to find one that best fit the data. The model types attempted can be seen in the table below.

ATM 2 Model Comparison Table (ETS and ARIMA)

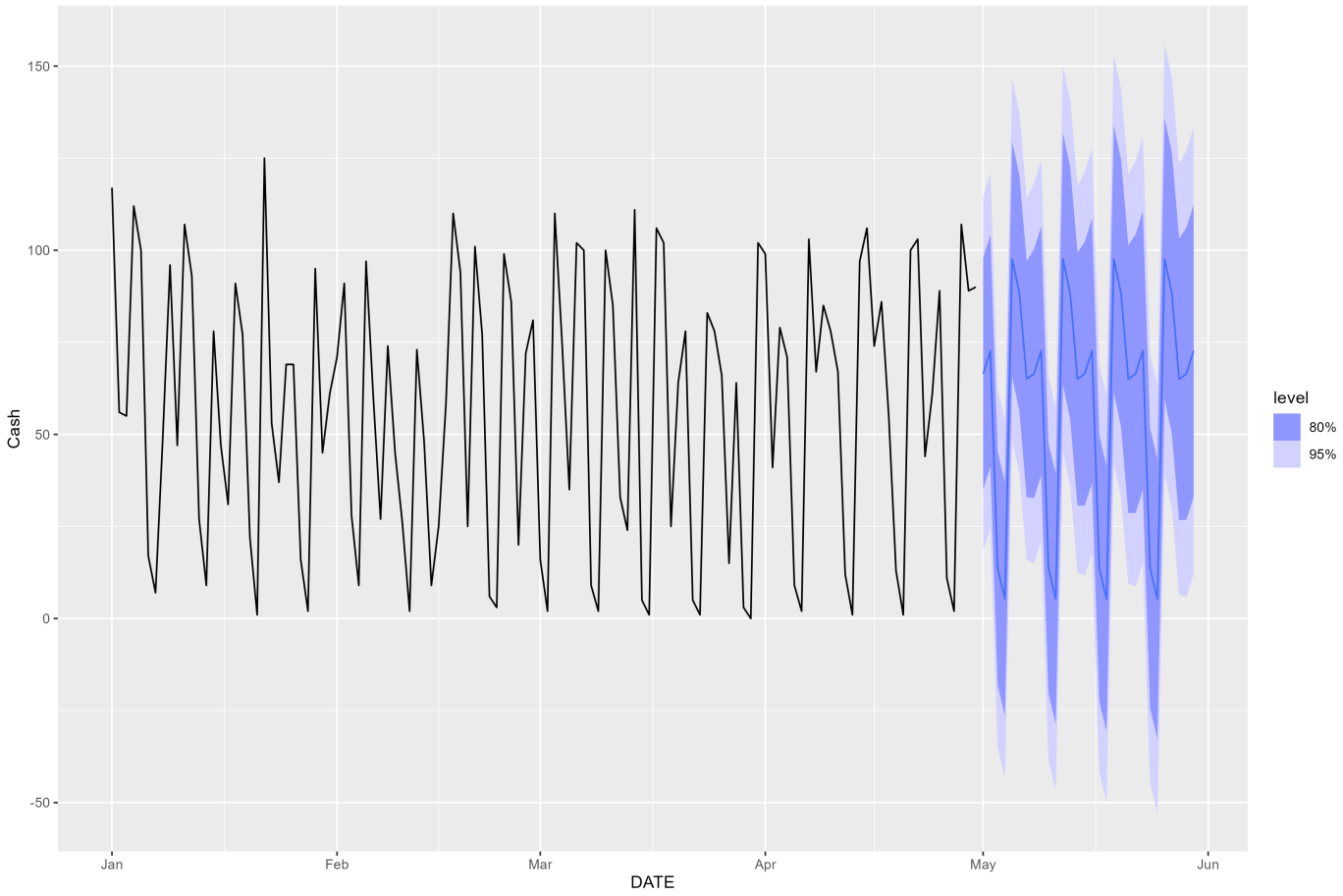
Model	Model Type	AIC	AICc	BIC	RMSE	Status
auto ETS(ANA)	ETS	4525.473	4526.095	4564.472	25.07654	Rejected
ANA	ETS	4525.473	4526.095	4564.472	25.07654	Rejected
manual_select2 ARIMA(0,0,0)(2,1,0)	ARIMA	3351.374	3351.441	3363.015	25.54244	Rejected
manual_select3 ARIMA(0,0,0)(3,1,0)	ARIMA	3339.141	3339.255	3354.664	25.01099	Rejected
auto_step ARIMA(2,0,2)(0,1,1)	ARIMA	3318.576	3318.816	3341.859	24.11392	Selected
auto_search ARIMA(2,0,2)(0,1,1)	ARIMA	3318.576	3318.816	3341.859	24.11392	Selected

After selecting the best model based on the numbers in the table, which was the ARIMA(2,0,2)(0,1,1) model, the residuals of the selected model were examined for good measure showing no autocorrelation and using

Ljung Box tests, the p-values confirmed this as well. I proceeded forward to forecast using this model, which can be seen in the image below.



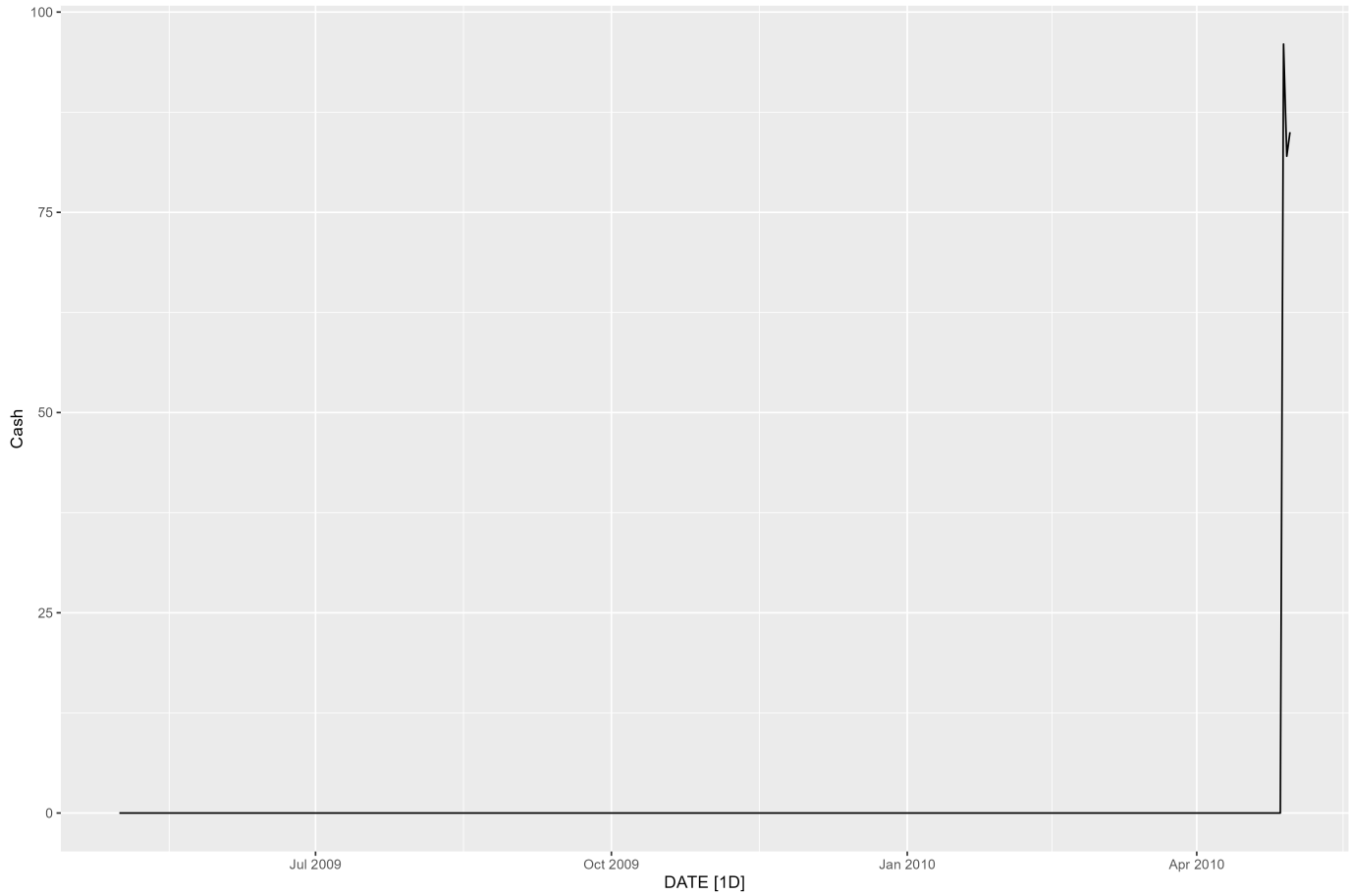
ATM2 Projection Forecast Plot



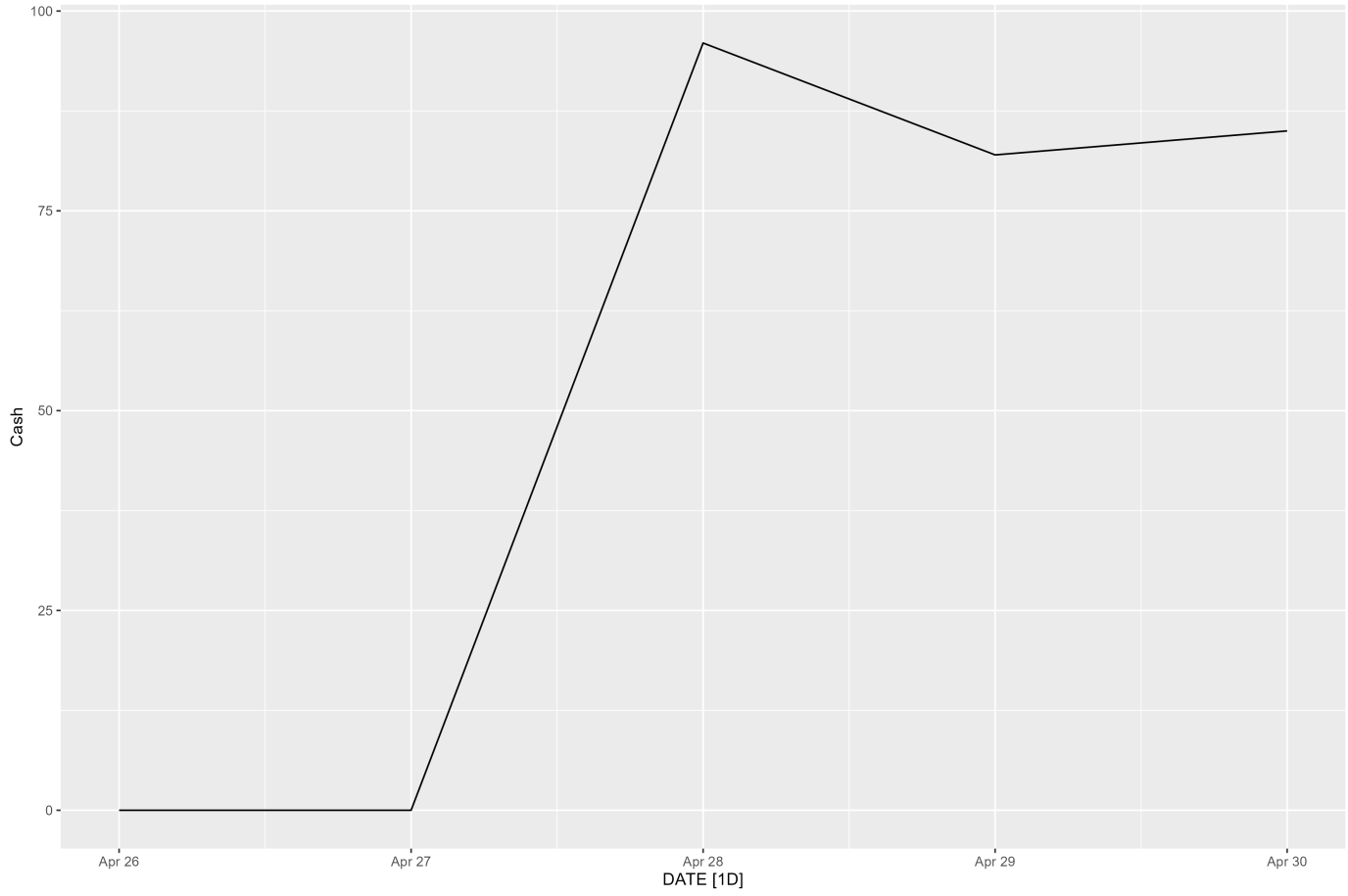
Overall, the forecast predicted that the Cash values for the month of May 2010, would average aroudn 6,000, or 60 (remember Cash is valued in hundreds). The forecast outlined that on the first of hte month the predicted values is around 66, by mid-month the forecast remains around 66, and by the end of the month the prediction is 72. There are variations in between. More details can be founf in the accompanying Excel file in the PARTA_ATM2 Tab. Lastly, all of my code and work for this ATM can be foudn in Appendix A.

ATM3

The following images are the plots of the raw data of ATM3. The data set was different as all but three of the data points for the Cash column had zero values. The last three points were the only varying data points available. This led to me using a different modeling technique than ATMs 1, 2 and 4.



Raw ATM3 Plot

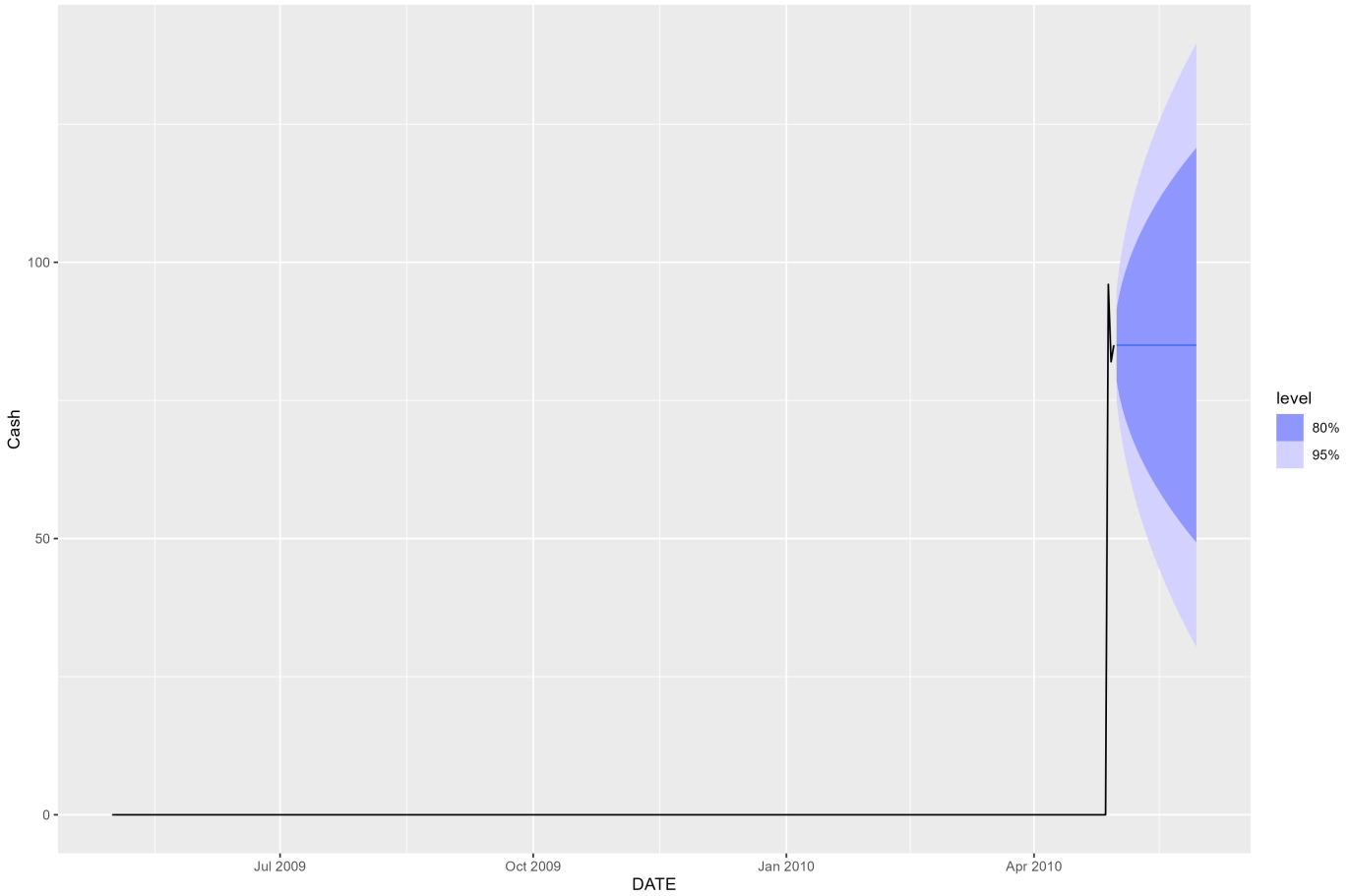


Raw ATM3 Plot Limted for Visualization

For modeling this data set ETS and ARIMA techniques were not used do it only have 3 non-zero data points towards the end of the timeframe covered by the data. Instead, I applied simple forecasting methods: Naive, Mean, and Drift models. After exampling the RMSE numbers, along with other measurements of error for the models I selected the NAIVE modeling method, and used this model to derive projected values for May 2010.

ATM 3 Model Comparison Table (Simple Models)

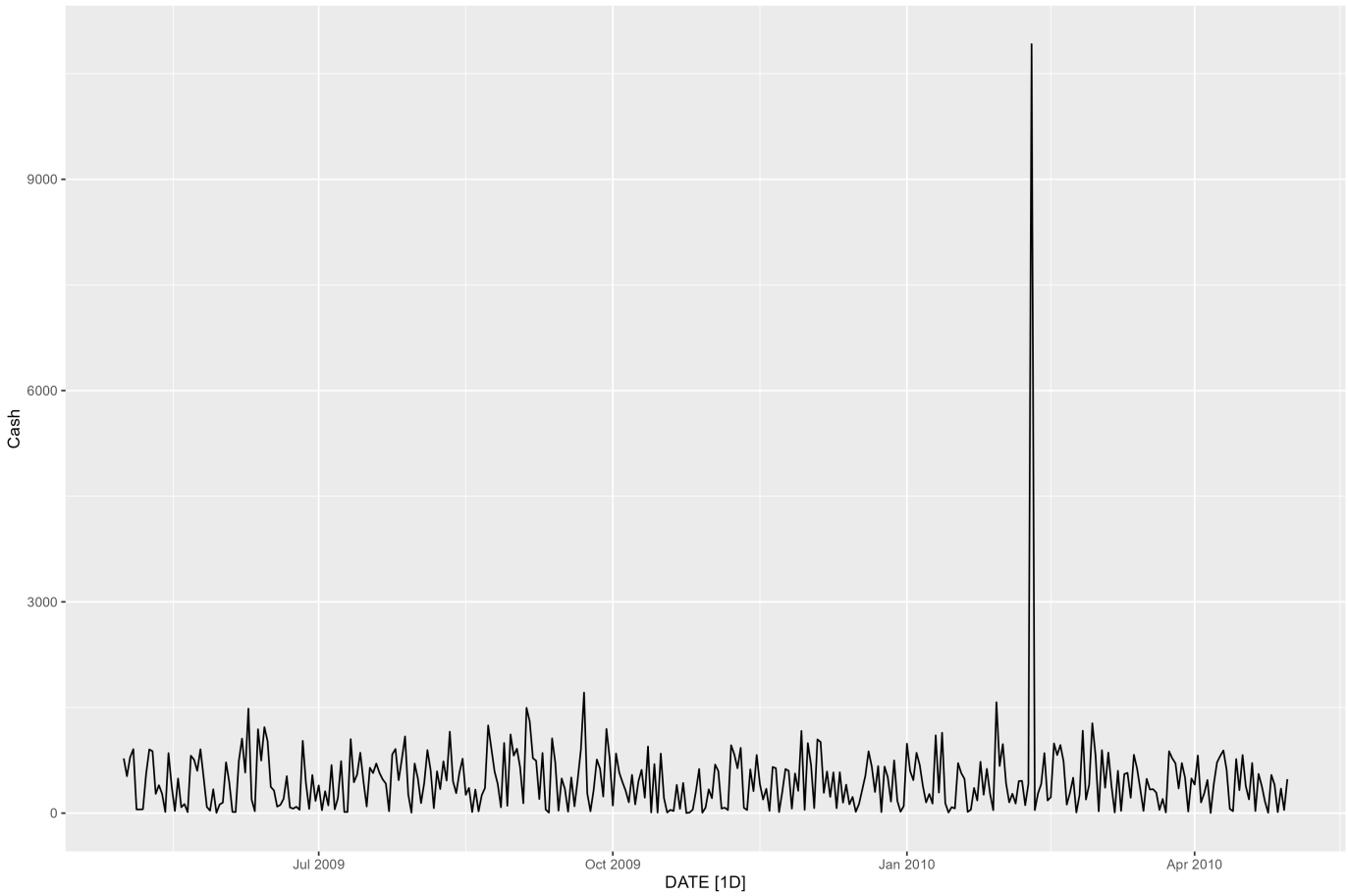
Model	RMSE	Notes
NAIVE	5.087423	Selected
MEAN	7.933887	Rejected
DRIFT	5.082060	Rejected



Overall, the forecast predicted that the values for May 2010 will be around 85. There is no variation in this model as it was the NAIVE modeling technique that was used due to the lack of nuanced data values. The data seems to show that while the ATM was deployed for a while, it is only recently that it began being used. More details can be found in the accompanying Excel file in the PARTA_ATM3 Tab. Lastly, like all predictions these are estimates, and the confidence intervals were withheld from the final output file for conciseness and clarity, but were included in the visualization. All of my code and work for this ATM can be found in Appendix A.

ATM4

The photo below shows the plotting of the raw ATM4 data.



Raw ATM4 Plot

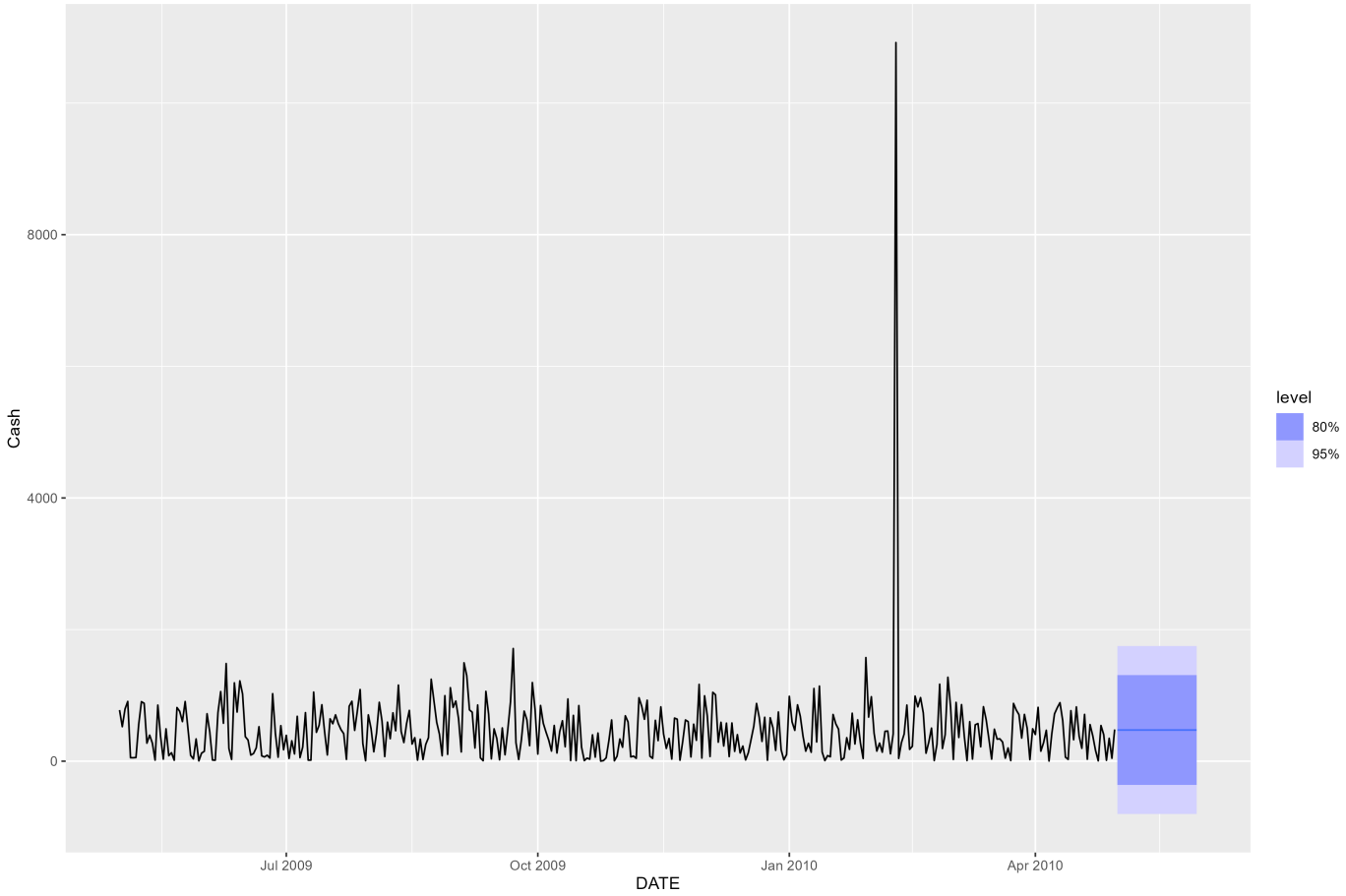
For ATM4, similar to ATM 1 and 2, the data has no clear trend. However, this data also no real pattern to the variations in the data. There was one large outlier that was in the data to note of, I left it because there was a sufficient amount of other data to model with. With this being said, the process for finding a model for ATM 4 was the same process for ATM1 and ATM2. I used ETS and ARIMA methodologies in order to find the best performing model to forecast May 2010 values.

ATM 4 Model Comparison Table (ETS and ARIMA)

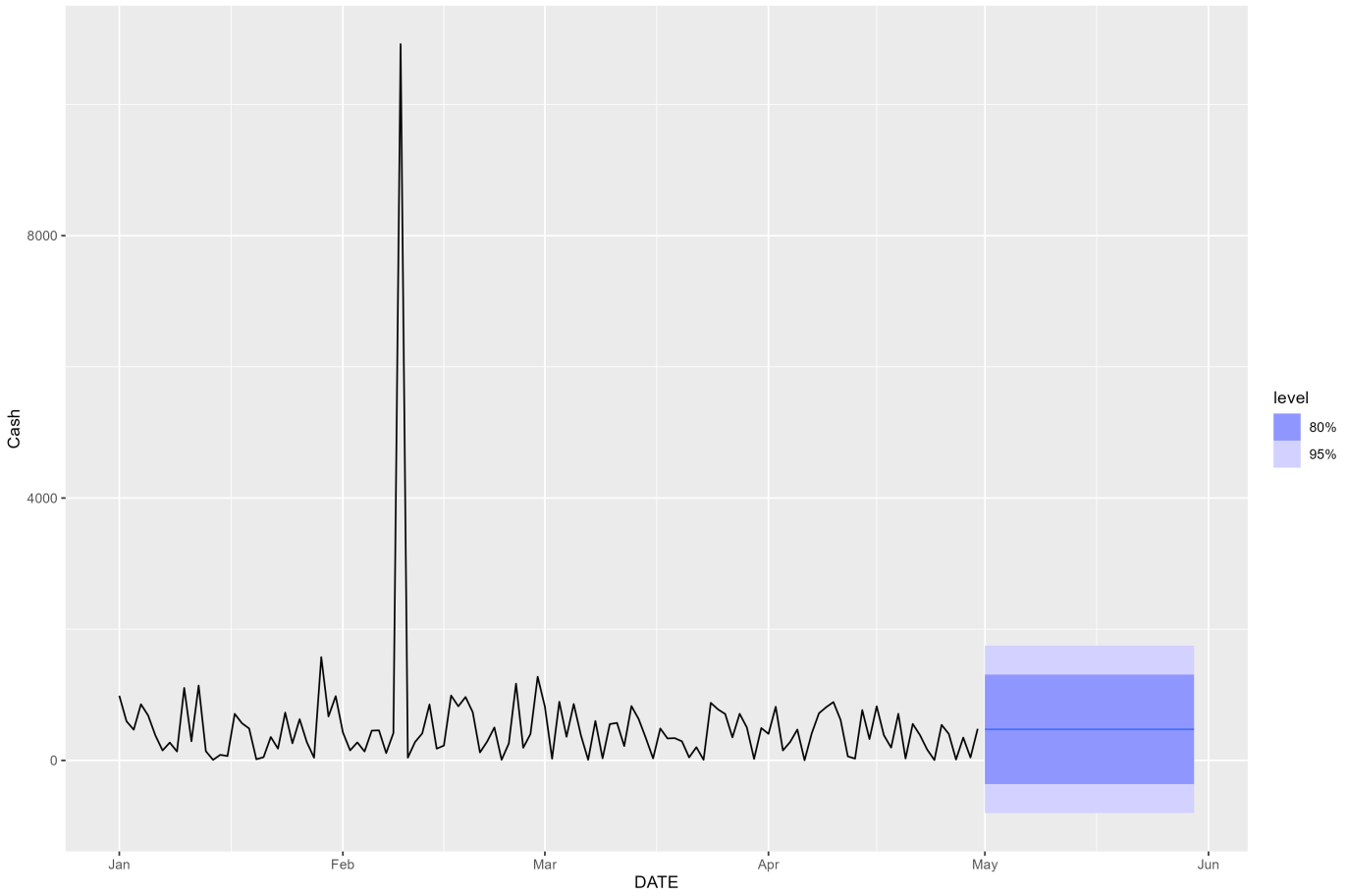
Model	Model Type	AIC	AICc	BIC	RMSE	Status
auto ETS(M,N,A)	ETS	6690.624	6691.246	6729.623	645.1182	Rejected
ANM	ETS	6884.976	6885.598	6923.975	635.3199	Rejected
MNM	ETS	6795.135	6795.756	6834.134	851.0784	Rejected
MNA	ETS	6690.624	6691.246	6729.623	645.1182	Rejected
manual_select2 ARIMA(0,0,0)(2,1,0)	ARIMA	5763.329	5763.397	5774.971	740.9888	Rejected
manual_select3 ARIMA(0,0,0)(3,1,0)	ARIMA	5736.843	5736.956	5752.365	710.4476	Rejected
auto_step ARIMA(0,0,0) w/ mean	ARIMA	5768.064	5768.097	5775.864	650.0437	Selected

Model	Model Type	AIC	AICc	BIC	RMSE	Status
auto_search ARIMA(0,0,0) w/ mean	ARIMA	5768.064	5768.097	5775.864	650.0437	Selected

After selecting the best model based on the numbers in the table, which was the ARIMA(0,0,0) w/ mean or basically a simple average of the data. The residuals of the selected model were examined for good measure showing odd patterns and using Ljung Box tests, the p-values confirmed no autocorrelation. I proceeded forward to forecast using this model, which can be seen in the image below.



ATM4 Projection Forecast Plot



Limited for Visibility ATM4 Projection Forecast Plot

Overall, the forecast predicted that the Cash value for the month of May in 2010 would be about 475 hundred. There is no variation in this data as it is a mean of the data itself. There is no weekly seasonlity in this data, and seems to just be random fluctuations on withdrawal usage. More details can be found in the accompanying Excel file in the PARTA_ATM4 Tab. Like all predictions these are estimates, and the confidence intervals were withheld from the final output file for conciseness and clarity, but were included in the visualization. Lastly, all of my code and work for this ATM can be found in Appendix A.

Part B – FORECASTING POWER (ResidentialCustomerForecastLoad-624.xlsx)

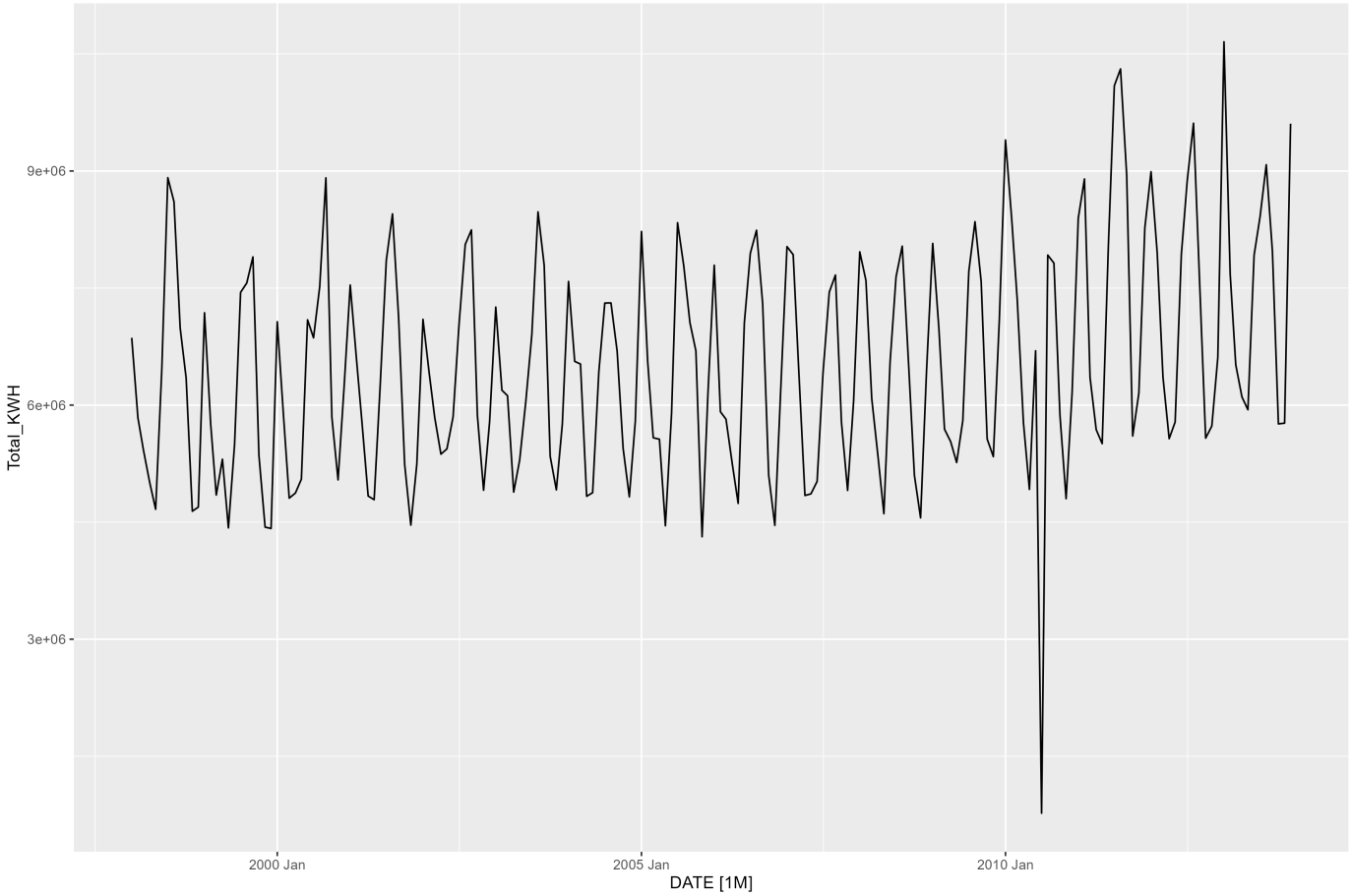
Description

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. Add this to your existing files above.

Overview

For Part B, the raw data was read into R, and processed as needed to prep for analysis. In order to prep the data, several steps were taken. Firstly, when initially read in the date values found in the “YYYY-MMM” column were not formatted as dates. This column was formatted as a date using yearmonth. It was after this that the

rest of the data cleaning could take place. The data was checked for null values that were in need of cleaning or imputation. There was one null found in the KWH values. Similar to Part A, because it was just one null compared to the larger data set, this value was filled in using the sandwiching KWH values. The one null was for September 2008, so the values of KWH for October 2008 and August 2008 were averaged to impute the September 2008 null value. After this was completed the data was placed into a tsibble, plotted and various models were applied ot the data. The image below shows the data plotted with the one imputed value.



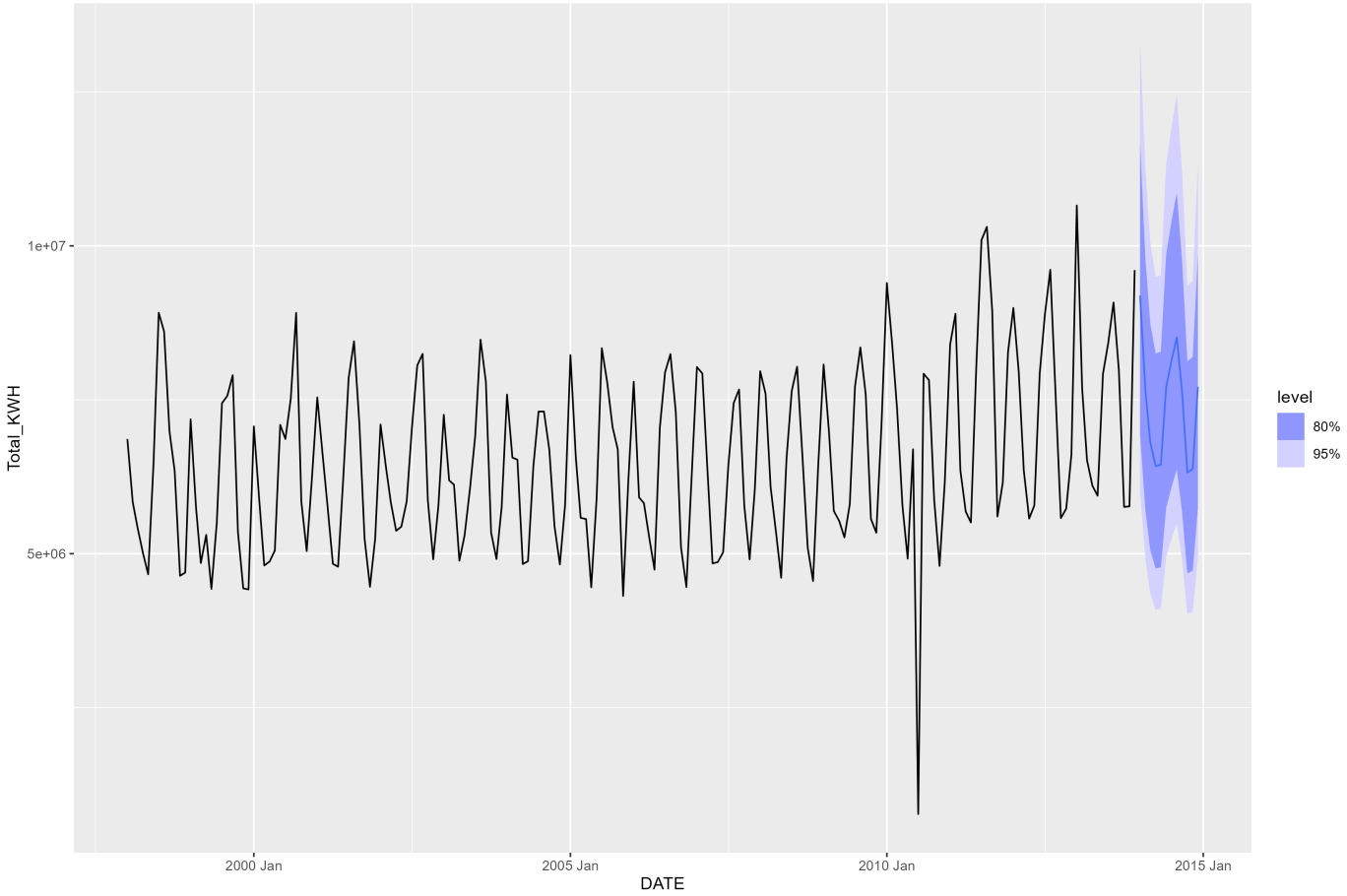
Part B KWH Data Plotted

Based on the experience in Part A, along with the patterns in the data, a Box-Cox lambda transformation value was used to best transform the data. The box-Cox yielded a value of 0.11, which is close to logging the data, as the most ideal transformation. After this, the ARIMA function was used to find the best fit model. Only ARIMA was used as in Part A ARIMA consistently performed better than the ETS models. The models that the function selected through both a step wise fashion, and the more time intensive method are in the table below.

Residential Power Model Comparison Table (ARIMA)

Model	Model Type	AIC	AICc	BIC	RMSE	Notes
auto_step ARIMA(0,1,2)(0,0,2)[12]	ARIMA	619.6074	619.9317	635.8688	1,197,100	
auto_search ARIMA(0,1,2)(2,0,0)[12]	ARIMA	601.4199	601.7443	617.6813	1,088,172	Selected model

After examining the output data for the models in the table above, the “auto_search” model was selected. I conducted one final round of checks on the model, confirming there was no autocorrelation or patterns in the models’ residuals by visually plotting them and performing a Ljung Box test to review the p-value. This model, the ARIMA(0,1,2)(2,0,0)[12], was used to then forecast the KWJ for the next year, or 12 months of 2014. THe forecast visualization can be seen plotted below.



PartB KWH Forecast for 2014

Overall, the forecast produced by the model outlines that in January 2014 the value for KWH will be ~9,192,398. By mid-year, the value will be the ~7,709,113, and by years end in December the value would be ~7,711,422. As with other ARIMA models there are variations in these values month to month. This means that power demand seems to peak in January 2014, decline until June where increases over the summer. Once fall comes power demand declined again, but slowly increases again in the winter months. Lastly, like all predictions these are estimates, and the confidence intervals were withheld from the final output file for conciseness and clarity, but were included in the visualization. All of my code and work for Part B can be found in Appendix B.

APPENDIX A - Part A Code & Analysis

```
# RAW FILE ALSO SITS HERE: https://raw.githubusercontent.com/jhnboyy/CUNY_SPS_WORK/main/Sprin
g2025/DATA624/Project1/ATM624Data.xlsx
#Reading in from local excel
atm_data <- read_excel("ATM624Data.xlsx")
print(head(atm_data))
```

```
## # A tibble: 6 × 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
```

```
print(nrow(atm_data))
```

```
## [1] 1474
```

```
## initial Processing notes:
# - Need to handle date, convert to actual date
# - Need to ensure the data is a time series.
```

Exploration and Cleaning

```
## Checking for nulls and other issues.
print(summary(atm_data))
```

```
##       DATE           ATM                Cash          
##  Min.   :39934   Length:1474        Min.   :    0.0  
##  1st Qu.:40026   Class :character   1st Qu.:    0.5  
##  Median :40118   Mode  :character   Median :   73.0  
##  Mean   :40118                      Mean   :  155.6  
##  3rd Qu.:40210                      3rd Qu.:  114.0  
##  Max.   :40312                      Max.   :10919.8  
##                                     NA's   :19
```

```
## The cash category has 19 nulls.
## Min of 0 max of 10,919. Mean of 155.
## Data has no nulls, right now is listed as number.
## ATM is character, non numeric. Looking at unique vals.

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

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

```
# There is an 'NA' Value here lets take a look

atm_null <- atm_data |> filter(is.na(ATM))

print(nrow(atm_null))
```

```
## [1] 14
```

```
# 14 rows where ATM is null.

print(unique(atm_null$Cash))
```

```
## [1] NA
```

```
# All Cash values are null for where the ATM column is null. These Rows should just be remove
d?

## Looking at the additional Cash values that are null with ATM values.
atm_data |> filter(!is.na(ATM), is.na(Cash))
```

```
## # A tibble: 5 × 3
##    DATE ATM    Cash
##   <dbl> <chr> <dbl>
## 1 39977 ATM1     NA
## 2 39980 ATM1     NA
## 3 39982 ATM2     NA
## 4 39986 ATM1     NA
## 5 39988 ATM2     NA
```

```
## Cash Nulls with ATM values are limited to ATM 1 and ATM 2. We should impute these 4 rows.

# Dropping the ATM_Nulls
atm_data <- atm_data |> filter(!is.na(ATM))
```

```
## After manually reviwing the data in excel data "DATE" column is actually a time stamp. How ever, its at midnight for each day so i think a date is good enough.
## Also had to google how excel does dates for the origin date. Evidently excel incorrectly t reats 1900 as a leap year when it wasnt. (INSANE!)
## So need to use 1899
#(Srouce: https://community.alteryx.com/t5/Alteryx-Designer-Desktop-Discussions/Rounding-Date Time-error-when-Importing-Excel/td-p/592023)
atm_data <-atm_data |> mutate(DATE = as.Date(DATE,origin = "1899-12-30"))

# Null Cash Imputation
#atm_data |> filter(is.na(Cash))

## Adding A column to Keep tabs on imputations
atm_data <- atm_data |>mutate(source = if_else(is.na(Cash), "imputed", "original"))

## Firstly Dealing with ATM 1 nulls
atm_data |> filter(is.na(Cash), ATM =='ATM1')
```

```
## # A tibble: 3 × 4
##   DATE       ATM    Cash source
##   <date>     <chr> <dbl> <chr>
## 1 2009-06-13 ATM1     NA imputed
## 2 2009-06-16 ATM1     NA imputed
## 3 2009-06-22 ATM1     NA imputed
```

```
# The nulls are limited 6/13/2009, 6/16/2009, 06/22/2009

## Checking the data for dates in this month for ATM1
atm_data |> filter(DATE>'2009-06-11',DATE<'2009-06-25' , ATM =='ATM1')
```

```
## # A tibble: 13 × 4
##    DATE       ATM    Cash source
##    <date>     <chr> <dbl> <chr>
##  1 2009-06-12 ATM1    142 original
##  2 2009-06-13 ATM1     NA imputed
##  3 2009-06-14 ATM1    120 original
##  4 2009-06-15 ATM1    106 original
##  5 2009-06-16 ATM1     NA imputed
##  6 2009-06-17 ATM1    108 original
##  7 2009-06-18 ATM1     21 original
##  8 2009-06-19 ATM1    140 original
##  9 2009-06-20 ATM1    110 original
## 10 2009-06-21 ATM1    115 original
## 11 2009-06-22 ATM1     NA imputed
## 12 2009-06-23 ATM1    108 original
## 13 2009-06-24 ATM1     66 original
```

```
## For ATM 1 there are values sandwiching each of the missing values so were going to just ta
ke the average of the "bookend" dates for each missing value. Its only 3 values.
#atm_data |> filter(DATE>'2009-06-11',DATE<'2009-06-15' , ATM =='ATM1') |> mutate()

atm_data_imputed <- atm_data |> mutate(Cash = if_else(is.na(Cash) & ATM == "ATM1" & DATE > a
s.Date("2009-06-11") & DATE < as.Date("2009-06-15"),
                                    (lag(Cash) + lead(Cash)) / 2, Cash ))
#Confirming
#atm_data_imputed |> filter(DATE>'2009-06-11',DATE<'2009-06-15' , ATM =='ATM1')

## Dealing with 6/16/2009 Null
#atm_data_imputed |> filter(DATE>'2009-06-14',DATE<'2009-06-18' , ATM =='ATM1')
atm_data_imputed <- atm_data_imputed |> mutate(Cash = if_else(is.na(Cash) & ATM == "ATM1" & D
ATE > as.Date("2009-06-14") & DATE < as.Date("2009-06-18"),
                                    (lag(Cash) + lead(Cash)) / 2, Cash ))
#Confirming
#atm_data_imputed |> filter(DATE>'2009-06-14',DATE<'2009-06-18' , ATM =='ATM1')

# Dealing with the 06/22/2009
atm_data_imputed <- atm_data_imputed |> mutate(Cash = if_else(is.na(Cash) & ATM == "ATM1" & D
ATE > as.Date("2009-06-20") & DATE < as.Date("2009-06-24"),
                                    (lag(Cash) + lead(Cash)) / 2, Cash ))

#atm_data_imputed |> filter(DATE>'2009-06-20',DATE<'2009-06-24' , ATM =='ATM1')

## FINISHED WITH ATM 1 NULLS, MOVING TO ATM2 NULLS
atm_data_imputed |> filter(is.na(Cash), ATM =='ATM2')
```

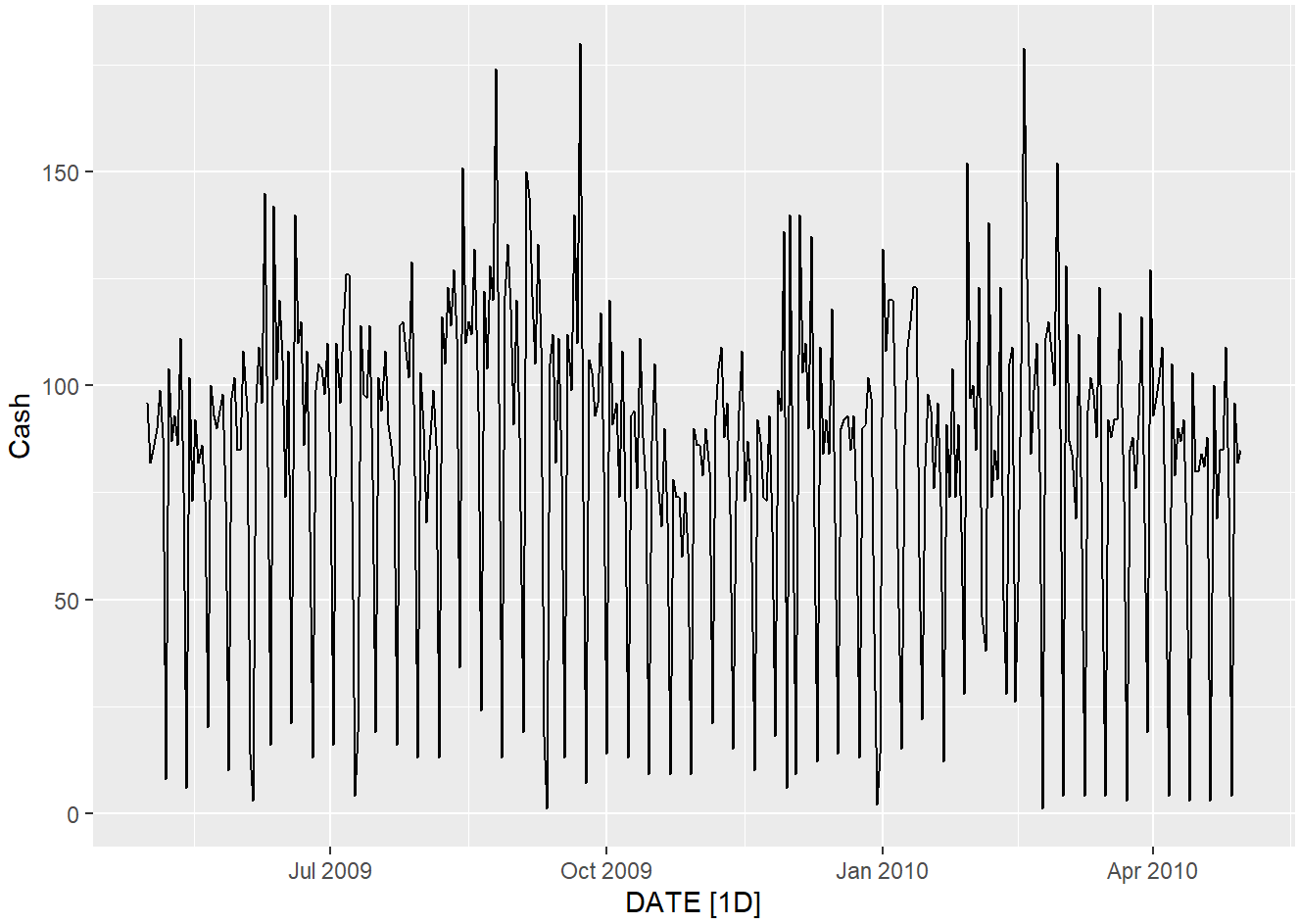
```
## # A tibble: 2 × 4
##   DATE       ATM    Cash source
##   <date>     <chr> <dbl> <chr>
## 1 2009-06-18 ATM2     NA imputed
## 2 2009-06-24 ATM2     NA imputed
```

```
## ATM 2 Nulls are limited to 06/18/2009 and 06/24/2009

## First instance at 6/18/2009
#atm_data_imputed |> filter(DATE>'2009-06-16',DATE<'2009-06-20' , ATM =='ATM2')
atm_data_imputed <- atm_data_imputed |> mutate(Cash = if_else(is.na(Cash) & ATM == "ATM2" & D
ATE > as.Date("2009-06-16") & DATE < as.Date("2009-06-20"),
                                  (lag(Cash) + lead(Cash)) / 2, Cash ))
#Checking
#atm_data_imputed |> filter(DATE>'2009-06-16',DATE<'2009-06-20' , ATM =='ATM2')

# Second instance at 6/24/2009
#atm_data_imputed |> filter(DATE>'2009-06-22',DATE<'2009-06-26' , ATM =='ATM2')
atm_data_imputed <- atm_data_imputed |> mutate(Cash = if_else(is.na(Cash) & ATM == "ATM2" & D
ATE > as.Date("2009-06-22") & DATE < as.Date("2009-06-26"),
                                  (lag(Cash) + lead(Cash)) / 2, Cash ))
#Checking
#atm_data_imputed |> filter(DATE>'2009-06-22',DATE<'2009-06-26' , ATM =='ATM2')

## Double checking there are no more nulls
#atm_data_imputed |> filter(is.na(Cash))

## Converting DF to a tsibble with no nulls
atm_tsibble <- atm_data_imputed |> as_tsibble(index = DATE, key = ATM)
#autoplot(atm_tsibble)
```

Preliminary Analysis

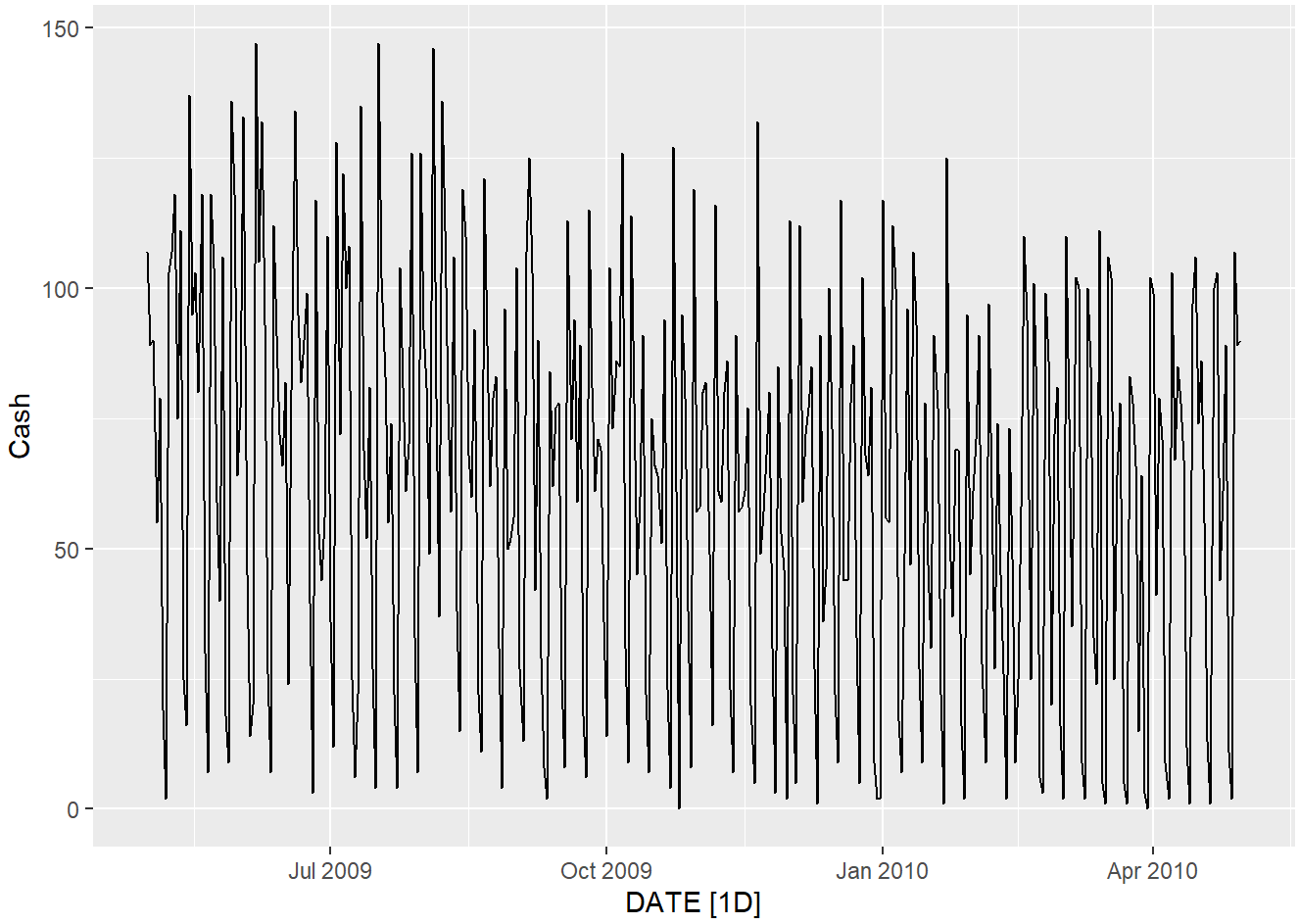
```
## Our goal is to parse out the cash (in hundreds) from each atm and give a projection for Ma
y 2010. Lets start to look at each atm.

## ATM1
atm1 <- atm_tsibble |> filter(ATM =='ATM1')
print(autoplot(atm1))
```

```
## Plot variable not specified, automatically selected `.vars = Cash`
```



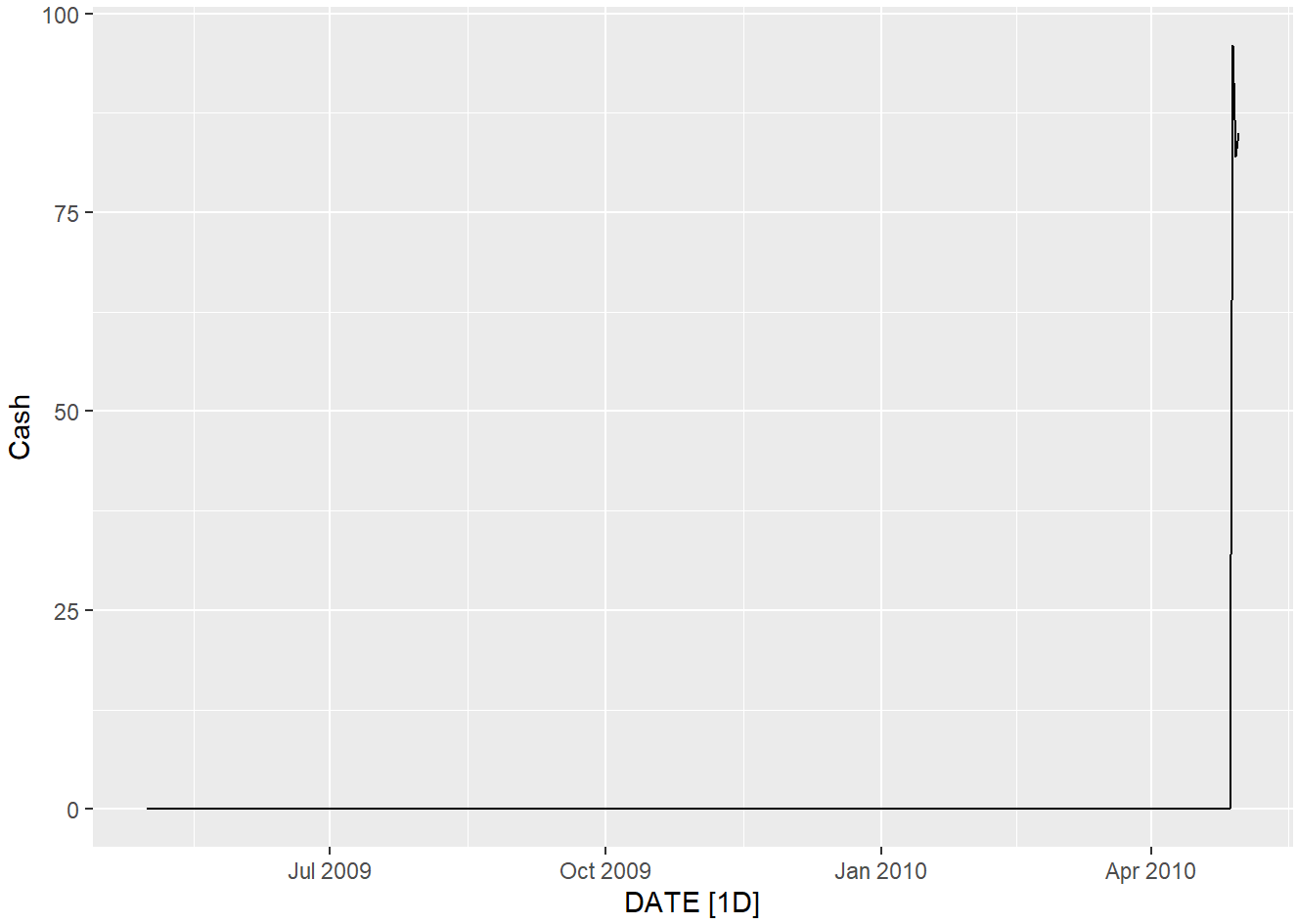

```
#p <- autoplot(atm1)
#ggsave("images/raw_atm1.png", plot = p, width = 12, height = 8, dpi = 300)

## Notes: Definitely seasonality here, but seemingly on a weekly or monthly time frame. Other than that variation there isnt really a trend, the data is fairly flat,

## ATM2
atm2 <- atm_tsibble |> filter(ATM =='ATM2')
print(autoplot(atm2))
```

```
## Plot variable not specified, automatically selected `.vars = Cash`
```



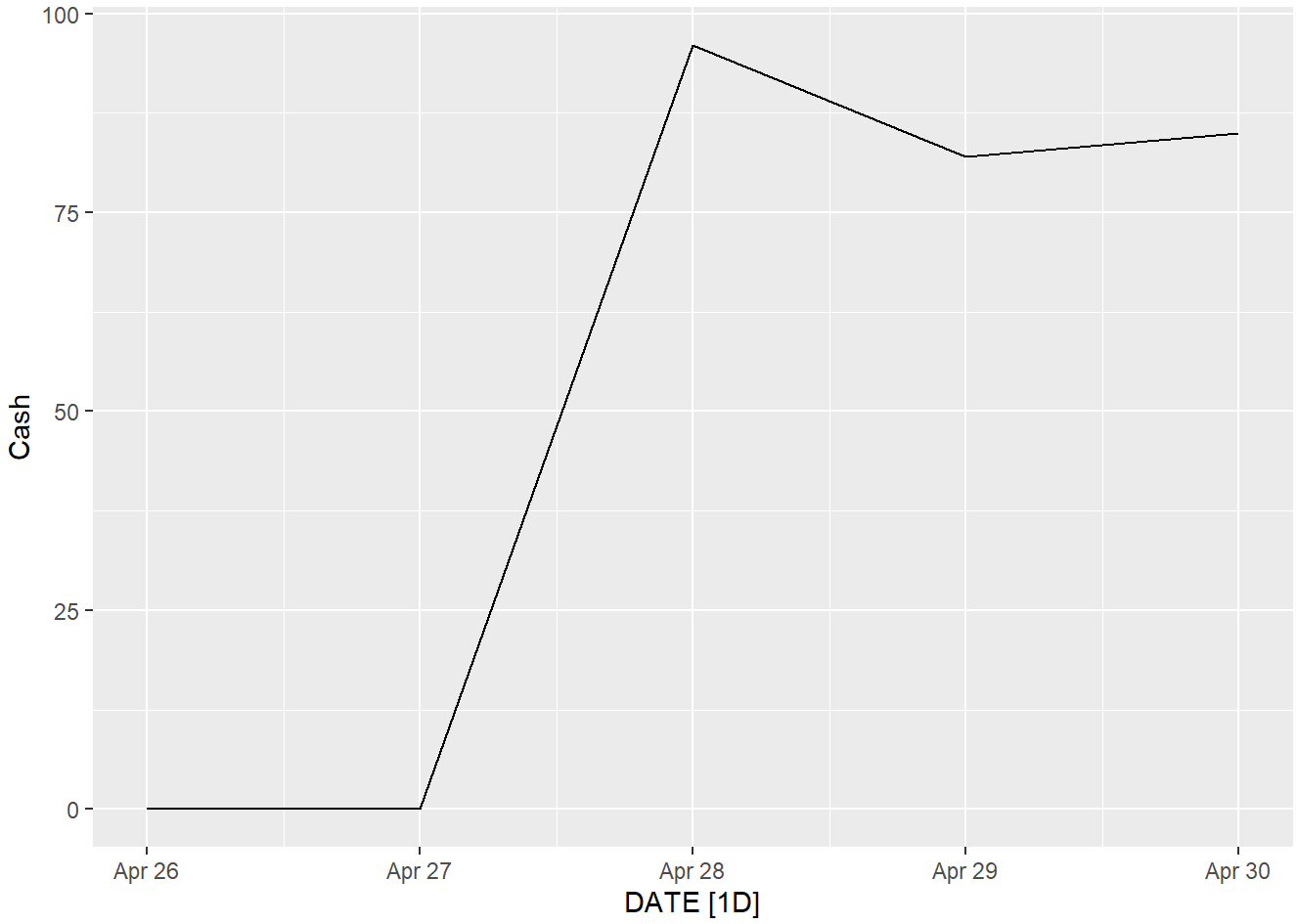

```
#p <- autoplot(atm2)
#ggsave("images/raw_atm2.png", plot = p, width = 12, height = 8, dpi = 300)
## Notes: Similar to ATM 1, no real trend, but there is definitely seasonality on a weekly or monthly timeframe.

## ATM 3
atm3 <- atm_tsibble |> filter(ATM =='ATM3')
print(autoplot(atm3))
```

```
## Plot variable not specified, automatically selected `.vars = Cash`
```



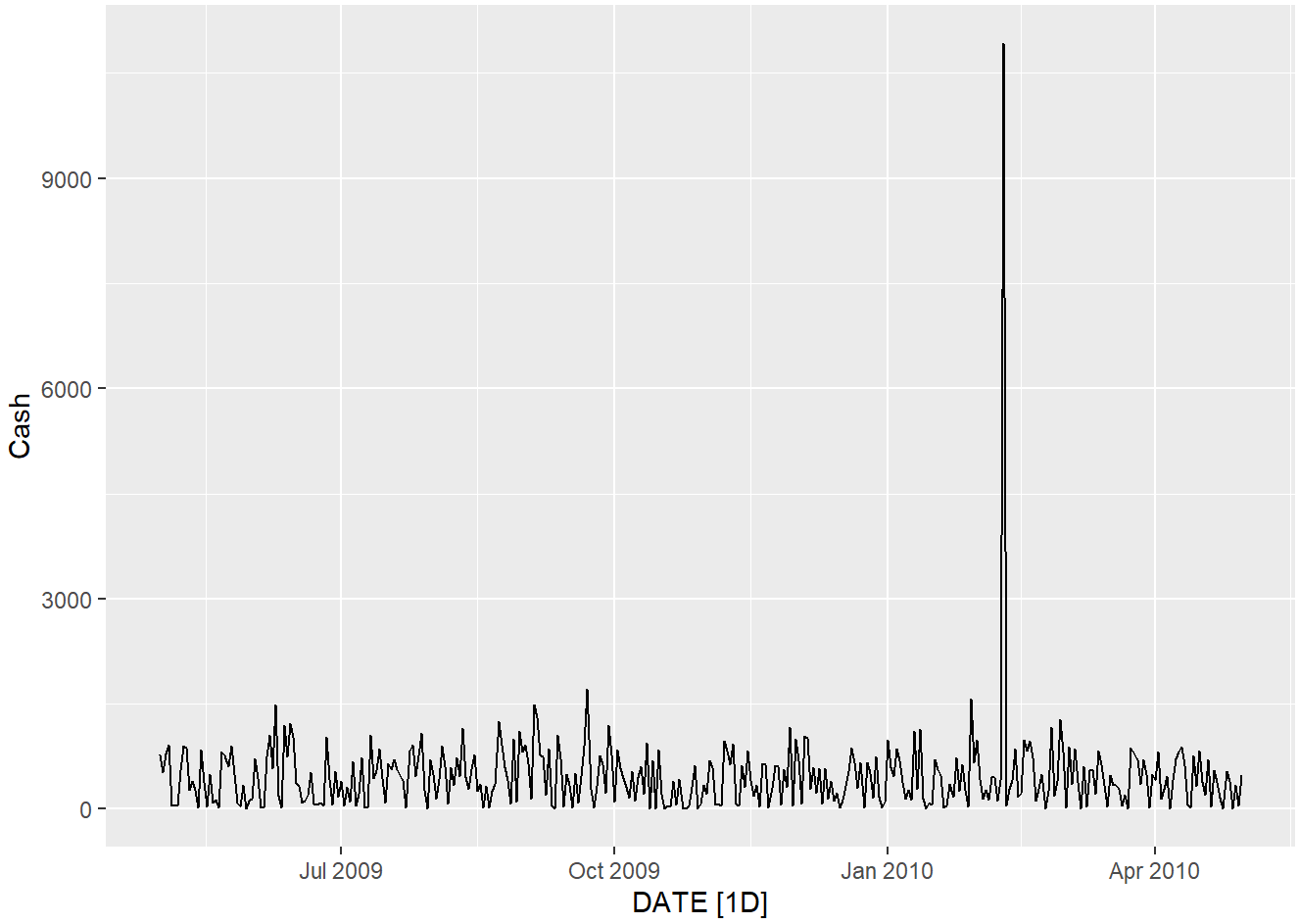

```
#p <- autoplot(atm3)
#ggsave("images/raw_atm3.png", plot = p, width = 12, height = 8, dpi = 300)
## NOTES: ATM 3 has 0 withdrawls for the vast majority of the timeframe looked at in the data
. Need to chop this chart down a bit to see the actual values.
## Data doesnt start until 4/28, in order to predict this ATM maybe we can use1,2, and 4?
lim_atm3 <- atm3 |> filter(DATE>=as.Date('2010-04-26'))
print(autoplot(lim_atm3))
```

```
## Plot variable not specified, automatically selected `.vars = Cash`
```




```
#p <- autoplot(atm3 |>  filter(DATE>=as.Date('2010-04-26')))
#ggsave("images/raw_atm3_lim.png", plot = p, width = 12, height = 8, dpi = 300)

## ATM 4
atm4 <- atm_tsibble |> filter(ATM =='ATM4')
print(autoplot(atm4))
```

```
## Plot variable not specified, automatically selected `.vars = Cash`
```




```
## notes: Plenty of data, there is one or 2 data points that are severe outliers and will inf
luence any forecast. Other than that there doesnt seem to be any patterns to the data. Also t
here is no trend in the data.
#p <- autoplot(atm4)
#ggsave("images/raw_atm4.png", plot = p, width = 12, height = 8, dpi = 300)

## Overall Take aways:
## - ATM 1, 2, 4 have no trend, but have seasonaility
## - ATM 3 has trend, but also limited to 3 data points. may need to estimate using other atm
s.
```

ATM1 Work

```
#atm1
#autoplot(atm1)

#Chcking for zeros
print(atm1 |> filter(Cash==0)) # No zeros, good for ETS
```

```
## # A tsibble: 0 x 4 [?]
## # Key:       ATM [0]
## # ℹ 4 variables: DATE <date>, ATM <chr>, Cash <dbl>, source <chr>
```

```
## Looking at ETS
auto_ets_atm1 <- atm1 |> model(auto_ANA = ETS(Cash), #ETS(A,N,A) is retrieved as the best aut
omatic fit.
                               ANM = ETS(Cash ~ error("A") + trend("N") + season("M")),
                               MNM = ETS(Cash ~ error("M") + trend("N") + season("M")),
                               MNA = ETS(Cash ~ error("M") + trend("N") + season("A")))


print(auto_ets_atm1 |> report())
```

```
## Warning in report.mdl_df(auto_ets_atm1): Model reporting is only supported for
## individual models, so a glance will be shown. To see the report for a specific
## model, use `select()` and `filter()` to identify a single model.
```

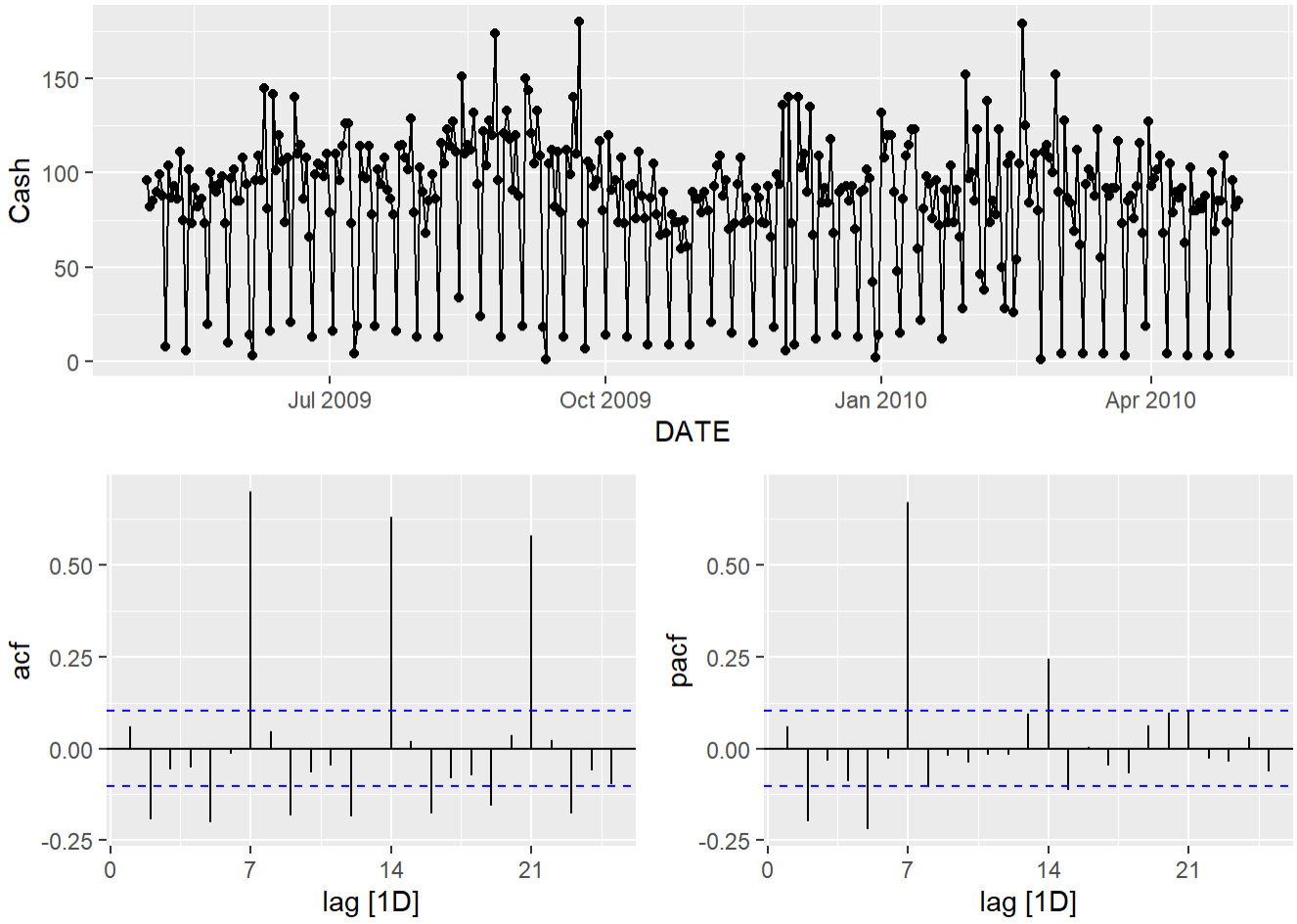
```
## # A tibble: 4 × 10
##   ATM   .model    sigma2 log_lik   AIC  AICc   BIC   MSE  AMSE    MAE
##   <chr> <chr>      <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ATM1  auto_ANA 582.     -2234. 4488. 4489. 4527.  568.  571. 15.1
## 2 ATM1  ANM      582.     -2234. 4488. 4489. 4527.  568.  570. 15.1
## 3 ATM1  MNM        0.134  -2273. 4567. 4567. 4606.  694.  708.  0.219
## 4 ATM1  MNA        0.148  -2288. 4595. 4596. 4634.  576.  577.  0.219
```

```
## Two best ETS models are ANM and the AUTO, the ANM is nearly the same, but a little better
with the AIC, AICc, and BIC models.

print(auto_ets_atm1 |> accuracy())
```

```
## # A tibble: 4 × 11
##   ATM   .model   .type         ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr> <chr>    <chr>      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ATM1  auto_ANA Training -0.0283  23.8  15.1 -106.  121. 0.849 0.854 0.129
## 2 ATM1  ANM      Training -0.449   23.8  15.1 -107.  121. 0.846 0.853 0.135
## 3 ATM1  MNM      Training -1.13    26.3  16.3 -128.  143. 0.913 0.943 0.0790
## 4 ATM1  MNA      Training  0.0644  24.0  15.4 -105.  120. 0.865 0.859 0.132
```

```
## The RMSE is basically the same for both, i think i may go with the ANM model as opposed to
the ANA auto model.

## Rerunning with just the good models.
auto_ets_atm1 <- atm1 |> model(auto_ANA = ETS(Cash), #ETS(A,N,A) is retrieved as the best aut
omatic fit.
                               ANM = ETS(Cash ~ error("A") + trend("N") + season("M")))


##   ------------------------------ Looking at ARIMA Models  ------------------------------

## Checking for Stationarity before modeling. No trend so may not need altering, or at least
a small amoutn of altering.

## Need to check stationarity
print(gg_tsdisplay(atm1,plot_type ="partial"))
```

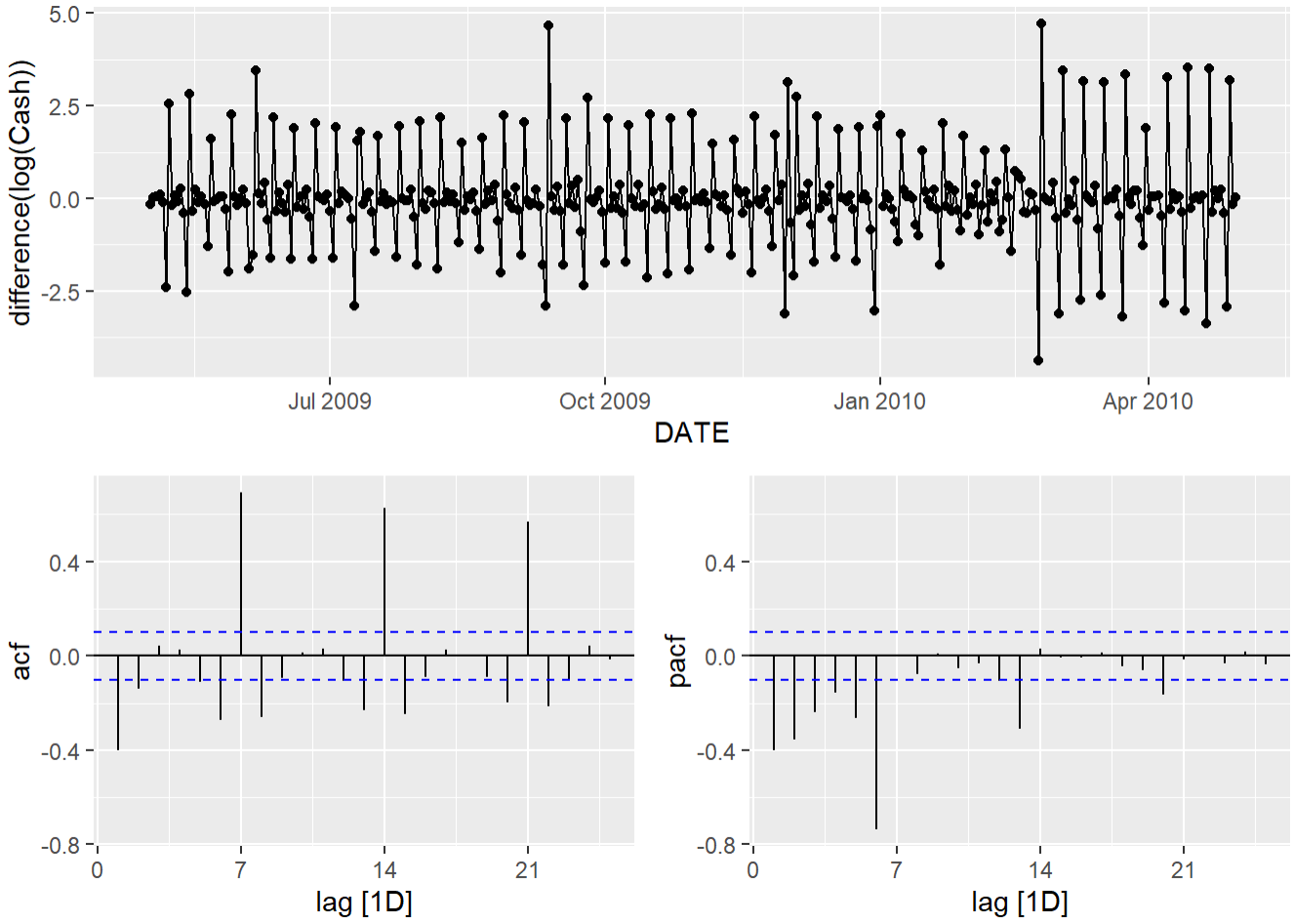
```
## Plot variable not specified, automatically selected `y = Cash`
```



```
# Could probably use a transformation. No zeros so just as log, no +1 needed.


print(atm1 |> gg_tsdisplay(difference(log(Cash)),plot_type ="partial"))
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```

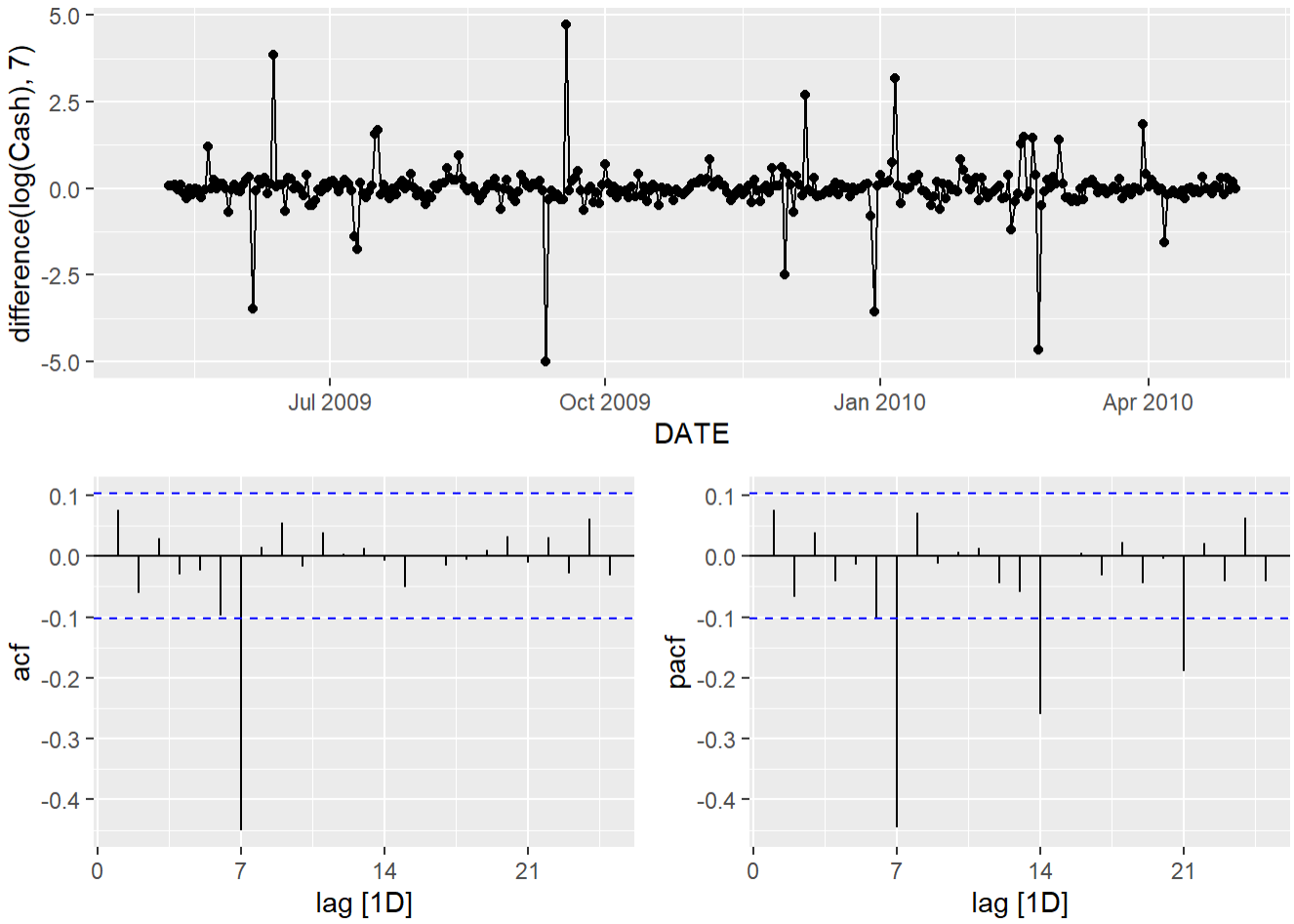
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# Looks much more stationary with log and one seasonal differencing at 7 for weekly seasonali
ty.
# ACF Shows some autocorrelation at week-level increments.
# PACF Shows similar. At increments of 6. So this may be AR if 2 or 3. No non-seasonal outlie
r. ARIMA(0,0,0)(2,1,0)[7] ?

#
print(atm1 |> gg_tsdisplay(difference(log(Cash),7),plot_type ="partial"))
```

```
## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
## SUing the PACF side more, there are no non seasonal oputliers, the seasonal outliers are 7
,14,21, so 2 or 3 for seasonal.

arima_atm1 <- atm1 |>
  model(manual_select2 = ARIMA(log(Cash) ~ pdq(0,0,0) + PDQ(2,1,0)),
        manual_select3 = ARIMA(log(Cash) ~ pdq(0,0,0) + PDQ(3,1,0)),
        auto_step = ARIMA(log(Cash)), #<ARIMA(0,0,0)(0,1,1)[7]>
        auto_search = ARIMA(log(Cash), stepwise = FALSE, approx=FALSE) )
arima_atm1
```

```
## # A mable: 1 x 5
## # Key:     ATM [1]
##   ATM             manual_select2           manual_select3
##   <chr>                  <model>                  <model>
## 1 ATM1  <ARIMA(0,0,0)(2,1,0)[7]> <ARIMA(0,0,0)(3,1,0)[7]>
## # ℹ 2 more variables: auto_step <model>, auto_search <model>
```

```
print(arima_atm1 |> report())
```

```
## Warning in report.mdl_df(arima_atm1): Model reporting is only supported for
## individual models, so a glance will be shown. To see the report for a specific
## model, use `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 4 × 9
##   ATM   .model         sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
##   <chr> <chr>           <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>
## 1 ATM1  manual_select2  0.374   -332.  670.  671.  682. <cpl [14]> <cpl [0]>
## 2 ATM1  manual_select3  0.363   -327.  661.  661.  677. <cpl [21]> <cpl [0]>
## 3 ATM1  auto_step       0.351   -322.  648.  648.  656. <cpl [0]>  <cpl [7]>
## 4 ATM1  auto_search     0.347   -319.  646.  646.  662. <cpl [0]>  <cpl [9]>
```

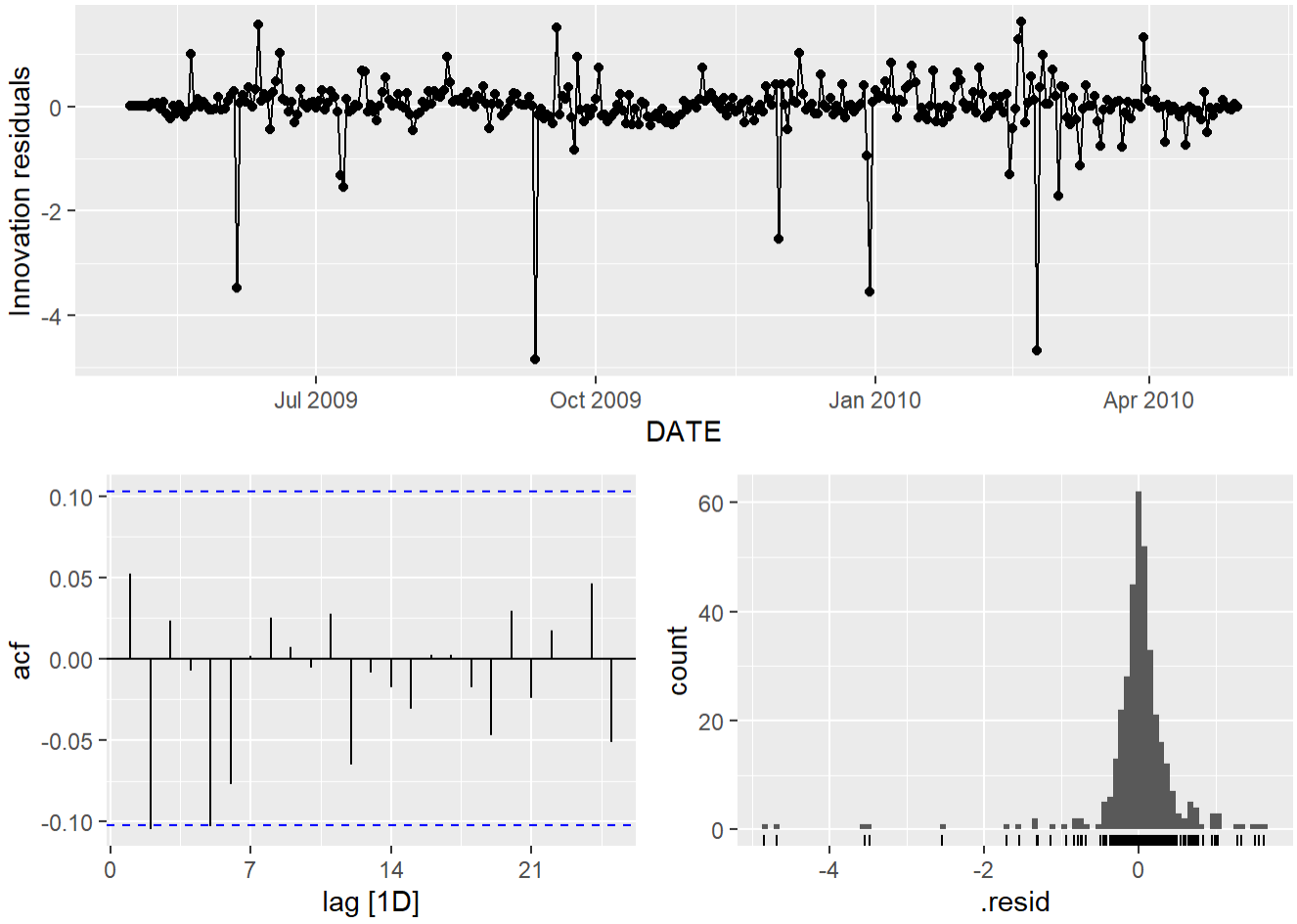
```
## Manually selected models are not as good as automated ones according to AIC, AICc and BIC.
I think auto_step model is the best, will be using that to forecast.

print(arima_atm1 |> accuracy())
```

```
## # A tibble: 4 × 11
##   ATM   .model         .type       ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr> <chr>          <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ATM1  manual_select2 Training  2.37  27.3  17.1 -91.0  112. 0.959 0.978 0.0915
## 2 ATM1  manual_select3 Training  2.88  26.8  16.9 -90.7  111. 0.950 0.958 0.0848
## 3 ATM1  auto_step      Training  3.66  26.0  16.4 -87.3  108. 0.924 0.931 0.112
## 4 ATM1  auto_search    Training  3.52  26.4  17.0 -84.0  105. 0.956 0.946 0.0294
```

```
## Confirming that auto_step is the best model. RMSE is 25.99


## COMPARING ETS AND ARIMA
### AUTOSTEP ARIMA (<ARIMA(0,0,0)(0,1,1)[7]>) -> 25.99424, AIC = 647.8779, AICc=    647.9117
, BIC= 655.6390
### ANM ETS -> RMSE=23.83080, AIC = 4488.277    , AICc= 4488.898, BIC=  4527.276


## OVERALL the RMSE is slightly better on the ANM model, but the AIC,AICc and BIC values for
the ARIMA are much better. Will be using ARIMA to forecast.
## Running again with only selected model
arima_atm1 <- atm1 |>
  model(auto_step = ARIMA(log(Cash)))

arima_atm1
```

```
## # A mable: 1 x 2
## # Key:     ATM [1]
##   ATM                 auto_step
##   <chr>                 <model>
## 1 ATM1  <ARIMA(0,0,0)(0,1,1)[7]>
```

```
## Last levels of confirmation checks
## looking at residuals
print(gg_tsresiduals(arima_atm1))
```




```
# Ljung Box test
print(augment(arima_atm1) |> features(.innov, ljung_box, lag=7, dof=1)) # Lag of 7 gets a pva
l of 0.07, above 0.05
```

```
## # A tibble: 1 × 4
##   ATM   .model    lb_stat lb_pvalue
##   <chr> <chr>       <dbl>     <dbl>
## 1 ATM1  auto_step    11.4    0.0761
```

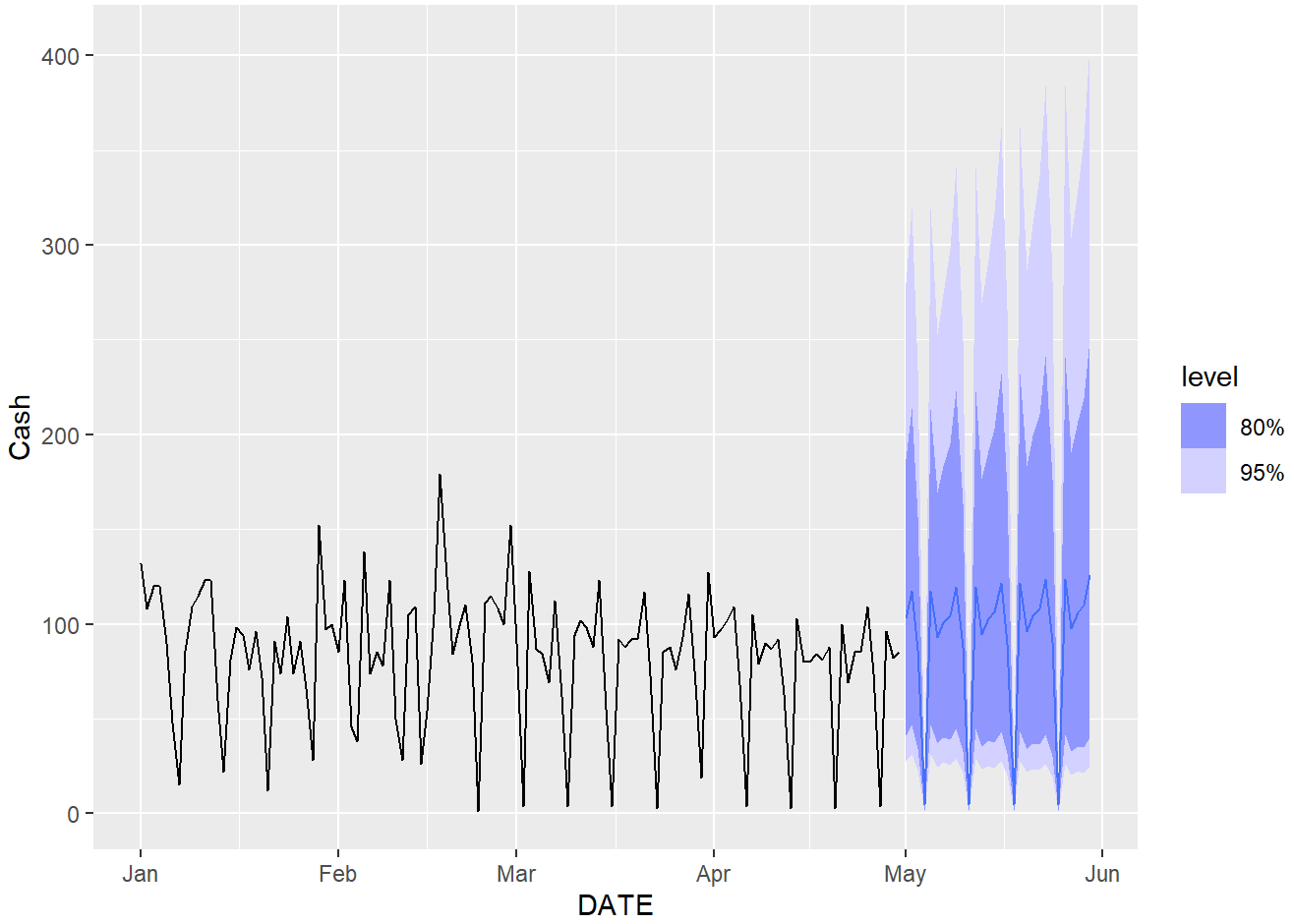
```
## Tryign with mulitples of 7
print(augment(arima_atm1) |> features(.innov, ljung_box, lag=14, dof=1)) #pval of 0.39
```

```
## # A tibble: 1 × 4
##   ATM   .model    lb_stat lb_pvalue
##   <chr> <chr>       <dbl>     <dbl>
## 1 ATM1  auto_step    13.7     0.393
```

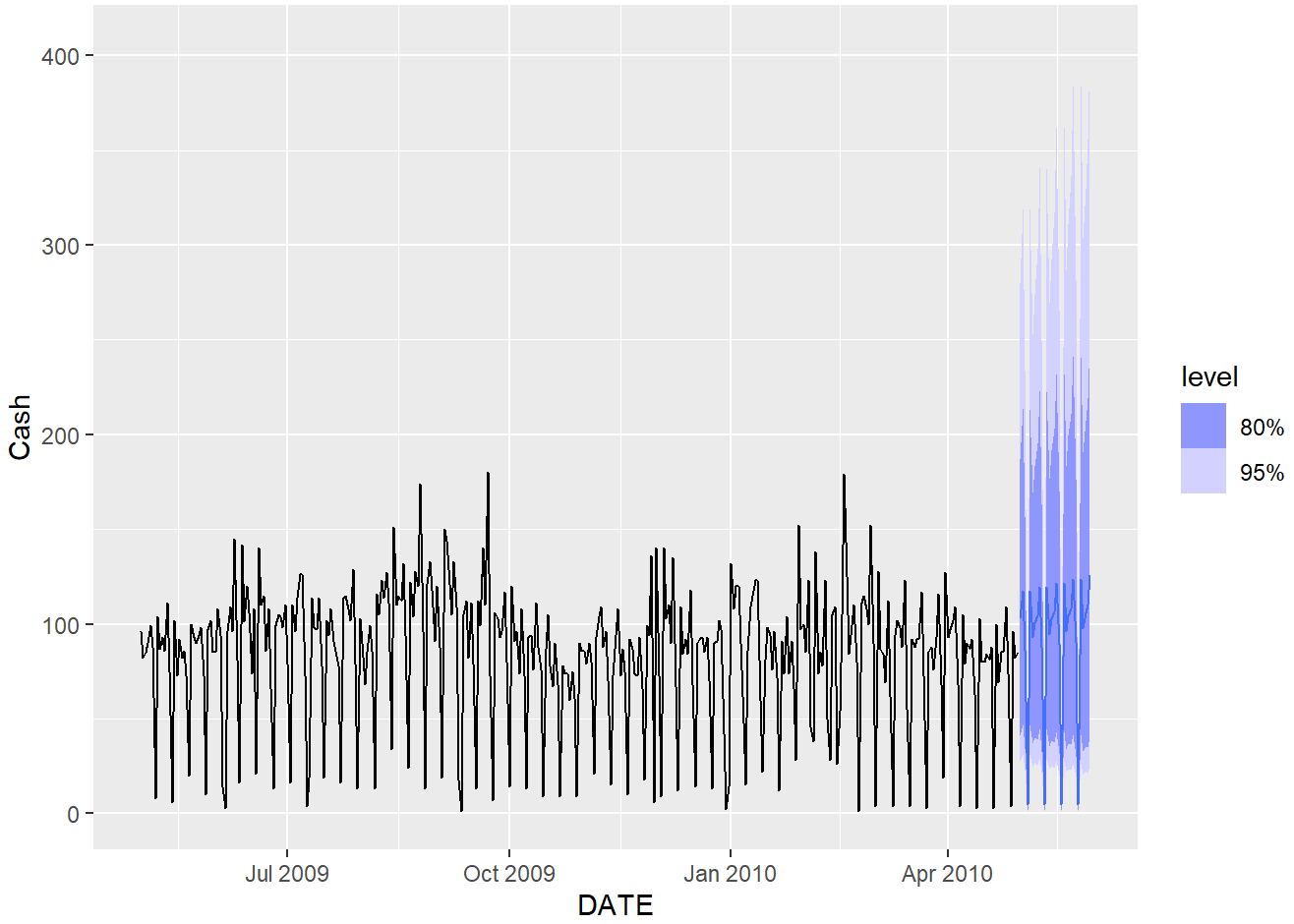
```
print(augment(arima_atm1) |> features(.innov, ljung_box, lag=21, dof=1)) #pval of 0.73
```

```
## # A tibble: 1 × 4
##   ATM   .model    lb_stat lb_pvalue
##   <chr> <chr>       <dbl>     <dbl>
## 1 ATM1  auto_step    15.6     0.739
```

```
# NO autocorrelation left in the residuals so its good. Movbing forward with forecast
arima_atm1 <- atm1 |>
  model(auto_step = ARIMA(log(Cash)))

### With model selected taking a look at the residuals for the
atm1_forecast <- arima_atm1 |>
  forecast(h = 30) #30 days

# ATM 1 forecast
print(atm1_forecast |>
    autoplot(atm1|>filter(DATE>=as.Date('2010-01-01'))))
```




```
#p <- atm1_forecast |>     autoplot(atm1|>filter(DATE>=as.Date('2010-01-01')))
#ggsave("images/atm1_proj_lim.png", plot = p, width = 12, height = 8, dpi = 300)

print(atm1_forecast |> autoplot(atm1))
```




```
#p <- atm1_forecast |>     autoplot(atm1)
#ggsave("images/atm1_proj_full.png", plot = p, width = 12, height = 8, dpi = 300)



## Looking at forecasted values
#atm1_forecast|> hilo() |> as_tsibble()

#forecast_table <- atm1_forecast |> hilo() |> as_tsibble()

# Write to CSV
#write.csv(atm1_forecast, "projection_data/atm1_forecast_values.csv", row.names = FALSE)
```

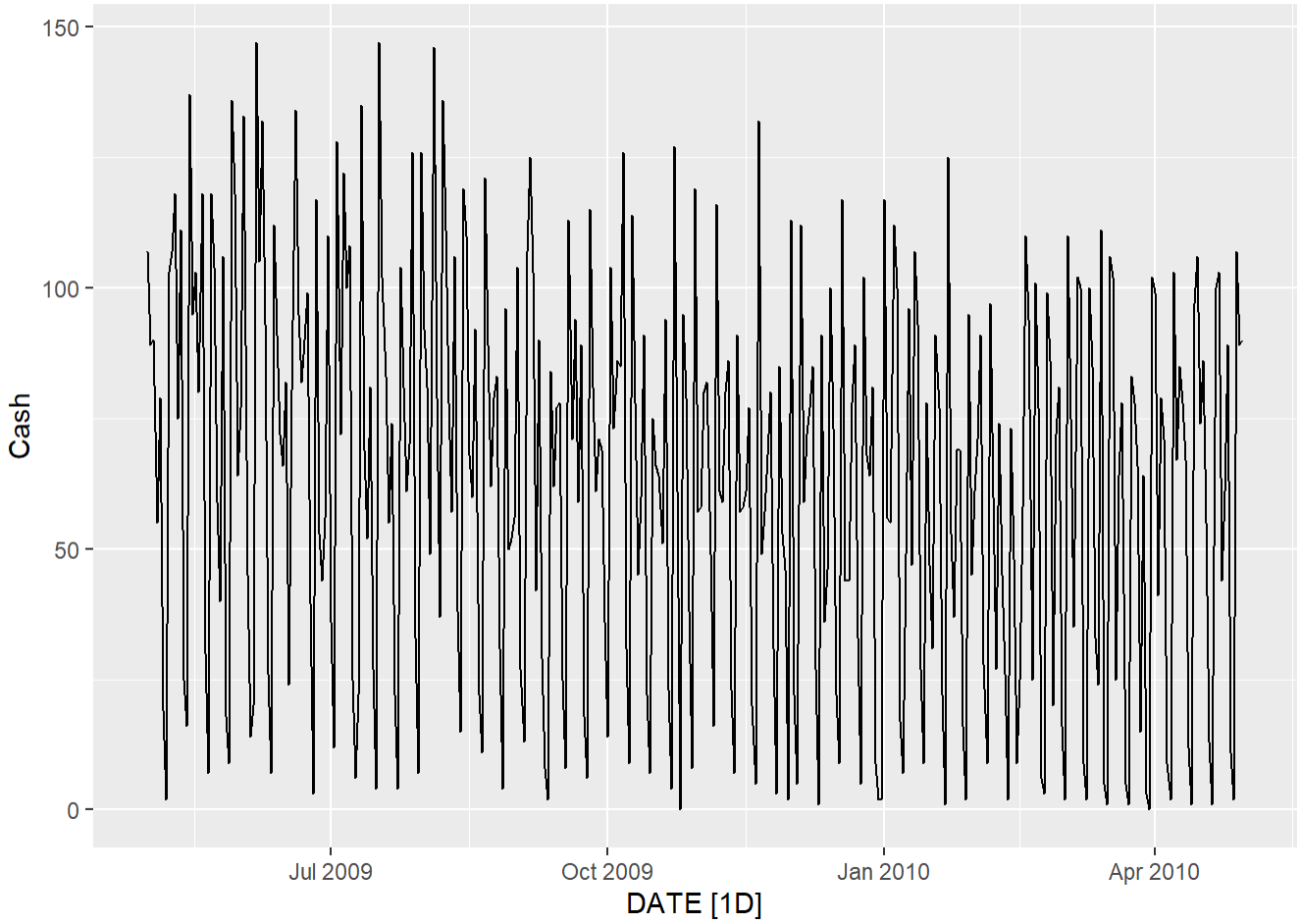
ATM2 Work

```
#Chcking for zeros
print(atm2 |> filter(Cash==0)) # Two zero values, would need to consider
```

```
## # A tsibble: 2 x 4 [1D]
## # Key:       ATM [1]
##   DATE       ATM    Cash source
##   <date>     <chr> <dbl> <chr>
## 1 2009-10-25 ATM2      0 original
## 2 2010-03-30 ATM2      0 original
```

```
autoplot(atm2)
```

```
## Plot variable not specified, automatically selected `.vars = Cash`
```




```
## Looking at ETS
auto_ets_atm2 <- atm2 |> model(auto = ETS(Cash),
                               ANA = ETS(Cash ~ error("A") + trend("N") + season("A")) #only
additive because of 0 values and no trend
                               )

print(auto_ets_atm2) #ETS(ANA) is retrieved as the best automatic fit.
```

```
## # A mable: 1 x 3
## # Key:     ATM [1]
##   ATM           auto          ANA
##   <chr>      <model>      <model>
## 1 ATM2  <ETS(A,N,A)> <ETS(A,N,A)>
```

```
## ANA is best ETS fit.

## Rerunning with just the good models.
auto_ets_atm2 <- atm2 |> model(auto_ANA = ETS(Cash))

print(auto_ets_atm2 |> report())
```

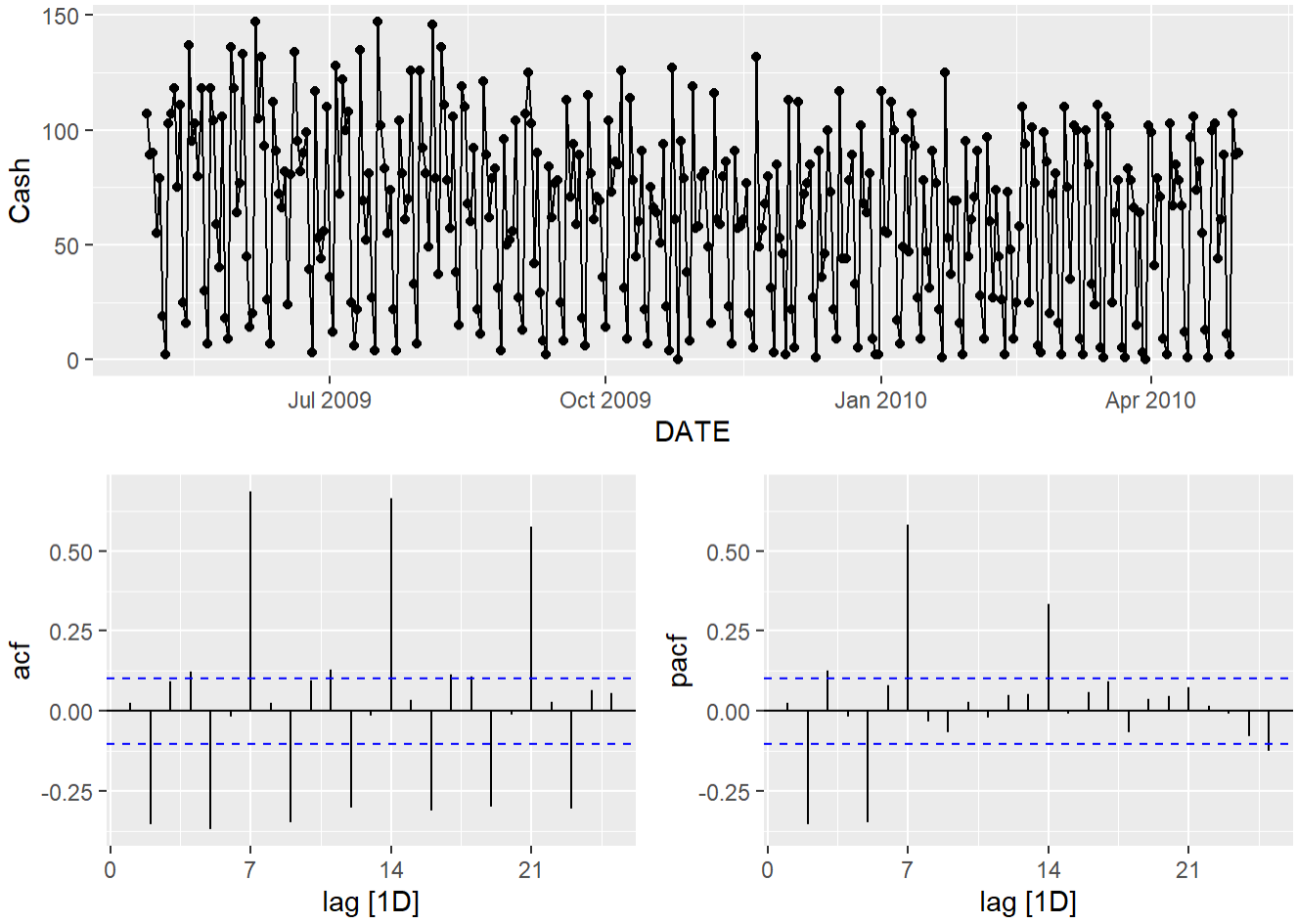
```
## Series: Cash
## Model: ETS(A,N,A)
##   Smoothing parameters:
##     alpha = 0.0001000394
##     gamma = 0.3586929
##
##   Initial states:
##      l[0]      s[0]    s[-1]   s[-2]     s[-3]    s[-4]    s[-5]   s[-6]
##  71.60993 -45.25081 -28.2608 9.83404 -3.416162 15.95563 32.60491 18.5332
##
##   sigma^2:  644.7302
##
##      AIC     AICc      BIC
## 4525.473 4526.095 4564.472
## # A mable: 1 x 2
## # Key:     ATM [1]
##   ATM       auto_ANA
##   <chr>      <model>
## 1 ATM2  <ETS(A,N,A)>
```

```
# AIC =4525.473, AICc= 4526.095, BIC=4564.472

print(auto_ets_atm2 |> accuracy())
```

```
## # A tibble: 1 × 11
##   ATM   .model   .type        ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr> <chr>    <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ATM2  auto_ANA Training -0.631  25.1  17.8  -Inf   Inf 0.857 0.831 0.0178
```

```
#RMSE 25.07654


##   ------------------------------ Looking at ARIMA Models  ------------------------------

## Checking for Stationarity before modeling. No trend so may not need altering, or at least
a small amount of altering.

print(gg_tsdisplay(atm2,plot_type ="partial"))
```

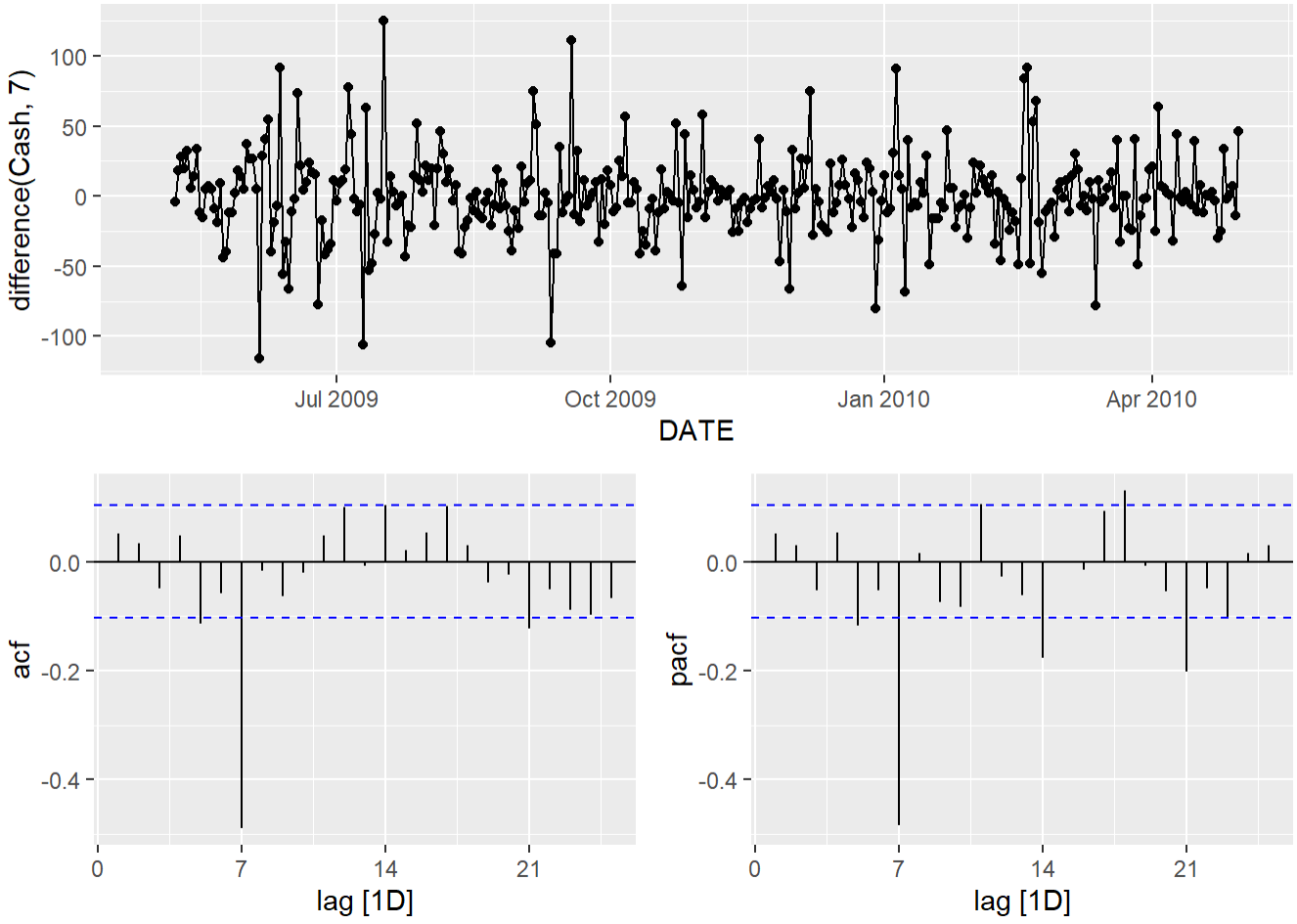
```
## Plot variable not specified, automatically selected `y = Cash`
```




```
# Mainly stationary, could probably use a transformation. Two zeros so log +1 woulb e needed
needed.


print(atm2 |> gg_tsdisplay(difference(Cash,7),plot_type ="partial"))
```

```
## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# Looks much more stationary with one seasonal differencing at 7 for weekly seasonality.
# ACF Shows some one outlier at 7
# PACF shows outliers  at 7 14 and 21. Biggest at 7. No non-seasonal outlier. ?


arima_atm2 <- atm2 |>
  model(manual_select2 = ARIMA(Cash ~ pdq(0,0,0) + PDQ(2,1,0)),
        manual_select3 = ARIMA(Cash ~ pdq(0,0,0) + PDQ(3,1,0)),
        auto_step = ARIMA(Cash), # SELECTED: <ARIMA(2,0,2)(0,1,1)[7]>

        auto_search = ARIMA(Cash, stepwise = FALSE, approx=FALSE) )# ALSO SELECTED <ARIMA
(2,0,2)(0,1,1)[7]>


#print(arima_atm2)
print(arima_atm2 |> report())
```

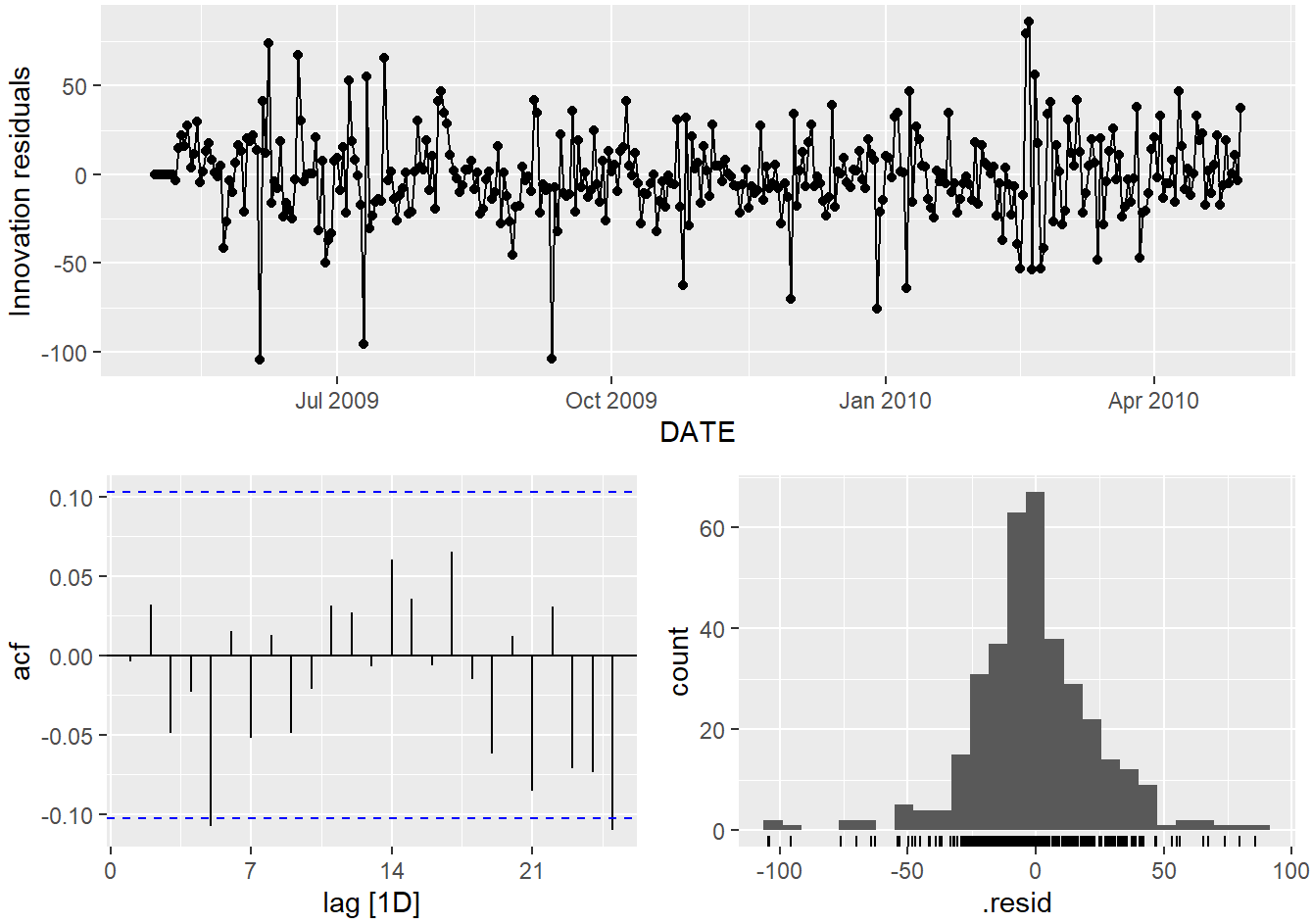
```
## Warning in report.mdl_df(arima_atm2): Model reporting is only supported for
## individual models, so a glance will be shown. To see the report for a specific
## model, use `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 4 × 9
##   ATM   .model         sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
##   <chr> <chr>           <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>
## 1 ATM2  manual_select2   669.  -1673. 3351. 3351. 3363. <cpl [14]> <cpl [0]>
## 2 ATM2  manual_select3   643.  -1666. 3339. 3339. 3355. <cpl [21]> <cpl [0]>
## 3 ATM2  auto_step        601.  -1653. 3319. 3319. 3342. <cpl [2]>  <cpl [9]>
## 4 ATM2  auto_search      601.  -1653. 3319. 3319. 3342. <cpl [2]>  <cpl [9]>
```

```
## Manually selected models are not as good as automated ones according to AIC, AICc and BIC.
I think auto models are the same, and better.
## <ARIMA(2,0,2)(0,1,1)[7]> AIC = 3318.576, AICc=3318.816, BIC=3341.859

print(arima_atm2 |> accuracy())
```

```
## # A tibble: 4 × 11
##   ATM   .model         .type     ME  RMSE   MAE   MPE  MAPE  MASE RMSSE     ACF1
##   <chr> <chr>          <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1 ATM2  manual_select2 Trai… -0.117  25.5  17.6  -Inf   Inf 0.848 0.846  0.0293
## 2 ATM2  manual_select3 Trai… -0.189  25.0  17.3  -Inf   Inf 0.833 0.829  0.00630
## 3 ATM2  auto_step      Trai… -0.891  24.1  17.0  -Inf   Inf 0.821 0.799 -0.00392
## 4 ATM2  auto_search    Trai… -0.891  24.1  17.0  -Inf   Inf 0.821 0.799 -0.00392
```

```
## Confirming that auto_step is the best model. RMSE is 24.11


## ---- COMPARING ETS AND ARIMA ----
### AUTOSTEP ARIMA -> 24.11, AIC = 3318.576, AICc=3318.816, BIC=3341.859
### ANA ETS -> RMSE=25.07654, AIC =4525.473, AICc= 4526.095, BIC=4564.472


## OVERALL the ARIMA numbers are much better. Will be using ARIMA to forecast.

## Running again with only selected model
arima_atm2 <- atm2 |>
  model(auto_step = ARIMA(Cash), # SELECTED: <ARIMA(2,0,2)(0,1,1)[7]>
        )


## Last levels of confirmation checks
## looking at residuals
print(gg_tsresiduals(arima_atm2))
```




```
# Ljung Box test
print(augment(arima_atm2) |> features(.innov, ljung_box, lag=7, dof=5)) # pval .031
```

```
## # A tibble: 1 × 4
##   ATM   .model    lb_stat lb_pvalue
##   <chr> <chr>       <dbl>     <dbl>
## 1 ATM2  auto_step    6.91    0.0316
```

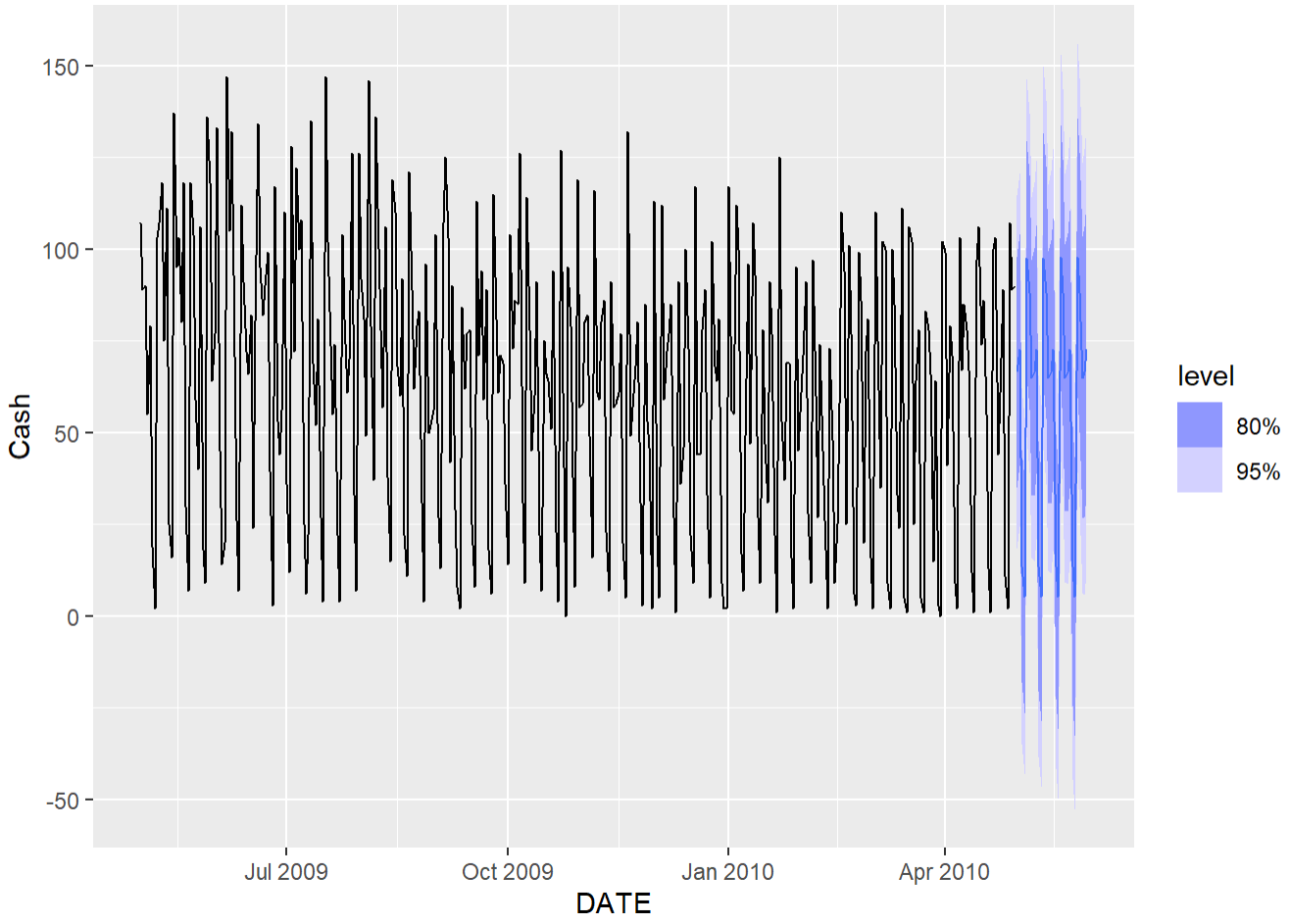
```
## Tryign with mulitples of 7
print(augment(arima_atm2) |> features(.innov, ljung_box, lag=14, dof=5)) #pval of 0.34
```

```
## # A tibble: 1 × 4
##   ATM   .model    lb_stat lb_pvalue
##   <chr> <chr>       <dbl>     <dbl>
## 1 ATM2  auto_step    10.1     0.345
```

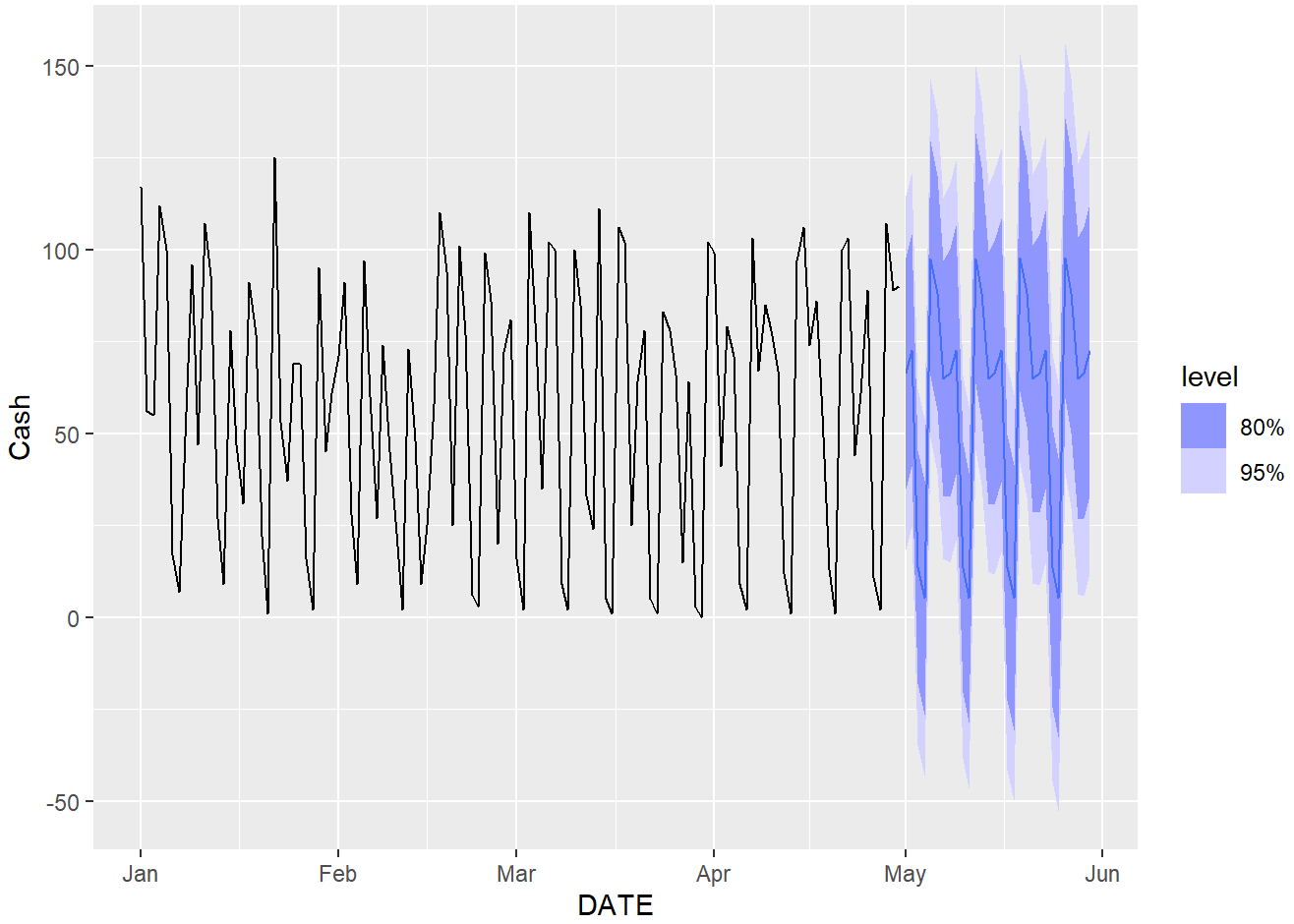
```
print(augment(arima_atm2) |> features(.innov, ljung_box, lag=21, dof=5)) #pval of 0.40
```

```
## # A tibble: 1 × 4
##   ATM   .model    lb_stat lb_pvalue
##   <chr> <chr>       <dbl>     <dbl>
## 1 ATM2  auto_step    16.7     0.406
```

```
# NO autocorrelation left in the residuals so its good. Moving forward with forecast
arima_atm2 <- atm2 |>
  model(auto_step = ARIMA(Cash))

### With model selected taking a look at the residuals for the
atm2_forecast <- arima_atm2 |>
  forecast(h = 30) #30 days

# ATM 2 forecast
print(atm2_forecast |>
    autoplot(atm2))
```




```
#p <- atm2_forecast |> autoplot(atm2)
#ggsave("images/atm2_proj.png", plot = p, width = 12, height = 8, dpi = 300)

print(atm2_forecast |>
    autoplot(atm2|>filter(DATE>=as.Date('2010-01-01'))))
```




```
#p <- atm2_forecast |> autoplot(atm2|>filter(DATE>=as.Date('2010-01-01')))
#ggsave("images/atm2_proj_lim.png", plot = p, width = 12, height = 8, dpi = 300)

## Looking at forecasted values
#atm2_forecast|> hilo() |> as_tsibble()

#forecast_table <- atm2_forecast |> hilo() |> as_tsibble()

# Write to CSV
#write.csv(atm2_forecast, "projection_data/atm2_forecast.csv", row.names = FALSE)
```

ATM3 Work

```
#Chcking for zeros
print(atm3 |> filter(Cash==0)) # 362 Zero values. this ATm Projection will need to be differe
nt from other 3
```

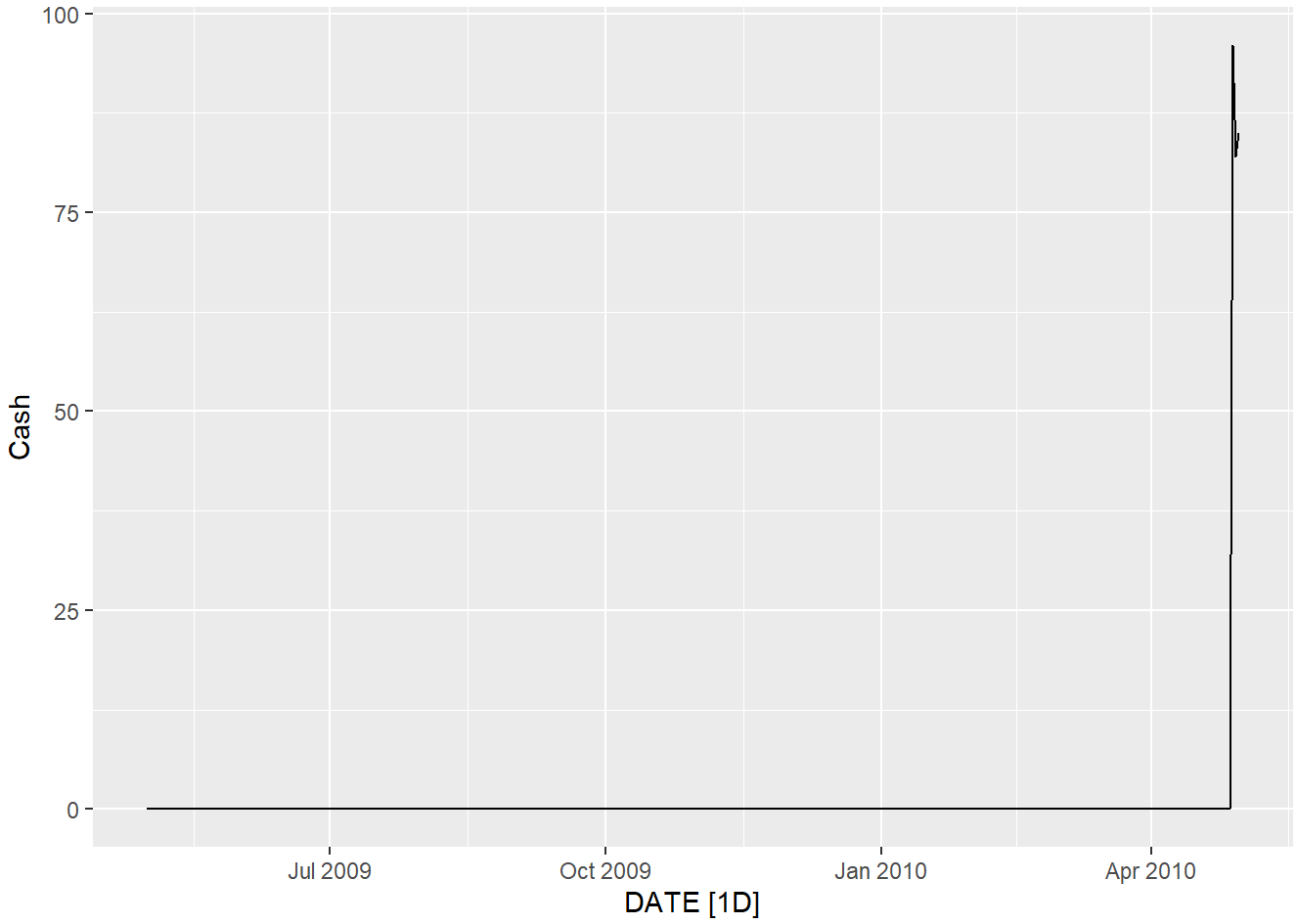
```
## # A tsibble: 362 x 4 [1D]
## # Key:       ATM [1]
##    DATE       ATM    Cash source
##    <date>     <chr> <dbl> <chr>
##  1 2009-05-01 ATM3      0 original
##  2 2009-05-02 ATM3      0 original
##  3 2009-05-03 ATM3      0 original
##  4 2009-05-04 ATM3      0 original
##  5 2009-05-05 ATM3      0 original
##  6 2009-05-06 ATM3      0 original
##  7 2009-05-07 ATM3      0 original
##  8 2009-05-08 ATM3      0 original
##  9 2009-05-09 ATM3      0 original
## 10 2009-05-10 ATM3      0 original
## # ℹ 352 more rows
```

```
print(atm3 |> filter(Cash!=0)) ## Only Three data points to use here. Different from the othe
r three projections.
```

```
## # A tsibble: 3 x 4 [1D]
## # Key:       ATM [1]
##   DATE       ATM    Cash source
##   <date>     <chr> <dbl> <chr>
## 1 2010-04-28 ATM3     96 original
## 2 2010-04-29 ATM3     82 original
## 3 2010-04-30 ATM3     85 original
```

```
autoplot(atm3)
```

```
## Plot variable not specified, automatically selected `.vars = Cash`
```




```
## Simple Modeling options Mean, NAIVe, and DRIFT
simple_models <- atm3 |> model(NAIVE = NAIVE(Cash),
              MEAN = MEAN(Cash),
              DRIFT = RW(Cash~drift()))

print(simple_models |> report())
```

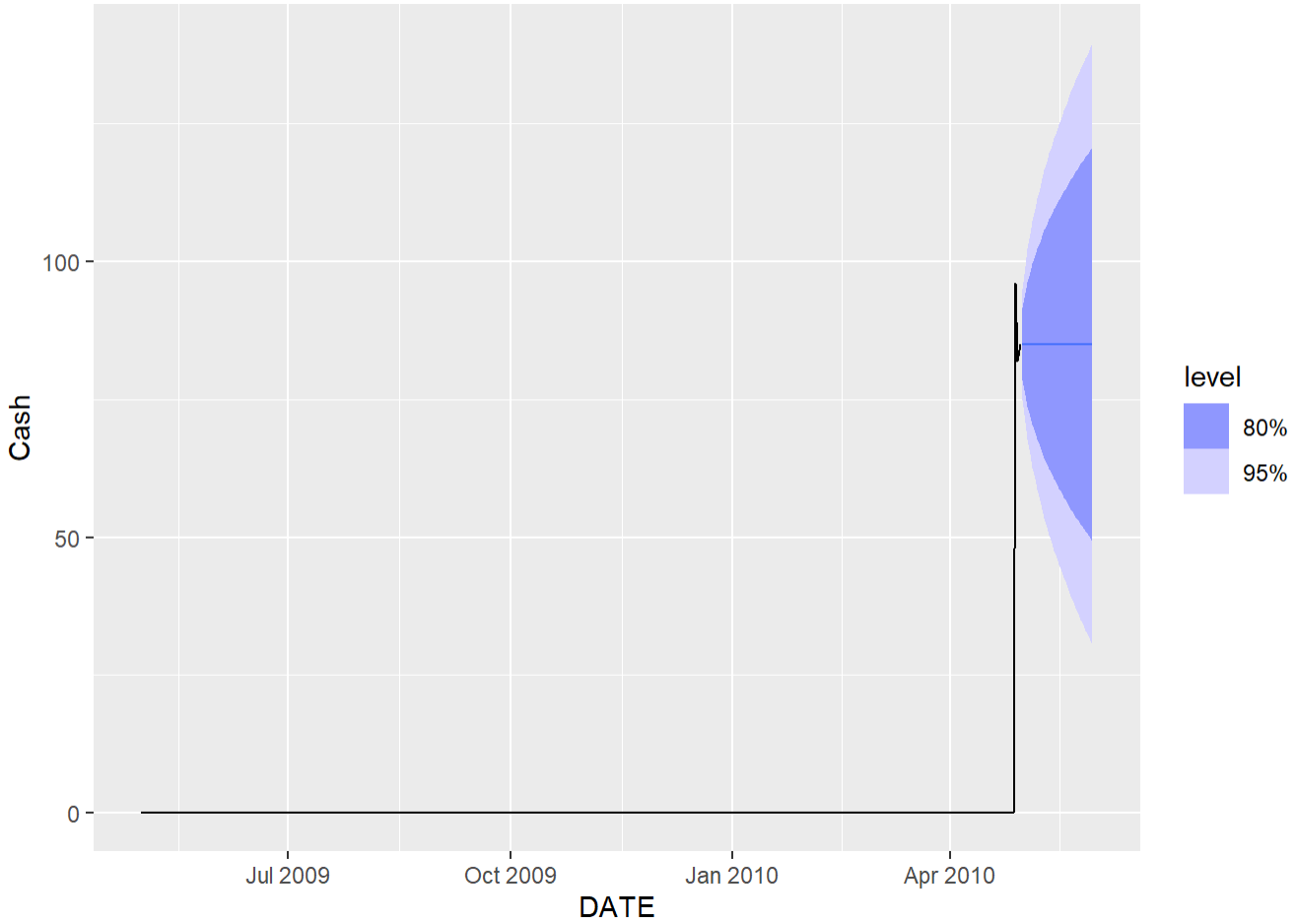
```
## Warning in report.mdl_df(simple_models): Model reporting is only supported for
## individual models, so a glance will be shown. To see the report for a specific
## model, use `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 3 × 3
##   ATM   .model sigma2
##   <chr> <chr>   <dbl>
## 1 ATM3  NAIVE    25.9
## 2 ATM3  MEAN     63.1
## 3 ATM3  DRIFT    25.9
```

```
print(simple_models |> accuracy())
```

```
## # A tibble: 3 × 11
##   ATM   .model .type           ME  RMSE   MAE    MPE  MAPE  MASE RMSSE   ACF1
##   <chr> <chr>  <chr>        <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ATM3  NAIVE  Training  2.34e- 1  5.09 0.310   28.8  40.2 0.423 0.632 -0.149
## 2 ATM3  MEAN   Training -5.63e-17  7.93 1.43  -Inf   Inf   1.95  0.986  0.640
## 3 ATM3  DRIFT  Training  1.09e-17  5.08 0.541 -Inf   Inf   0.737 0.632 -0.149
```

```
# The error measures seem tobe the lowest fro the NAIVE model, so I will use this for the for
ecast.

## Doing again with just NAIVE
simple_models <- atm3 |> model(NAIVE = NAIVE(Cash))

### With model selected taking a look at the residuals for the
atm3_forecast <- simple_models |> forecast(h = 30) #30 days

# ATM 3 forecast
print(atm3_forecast |>
    autoplot(atm3))
```



```
#p <- atm3_forecast |> autoplot(atm3)
#ggsave("images/atm3_proj.png", plot = p, width = 12, height = 8, dpi = 300)

## Looking at forecasted values
#atm3_forecast#|> hilo() |> as_tsibble()

#write.csv(atm3_forecast, "projection_data/atm3_forecast_values.csv", row.names = FALSE)
```

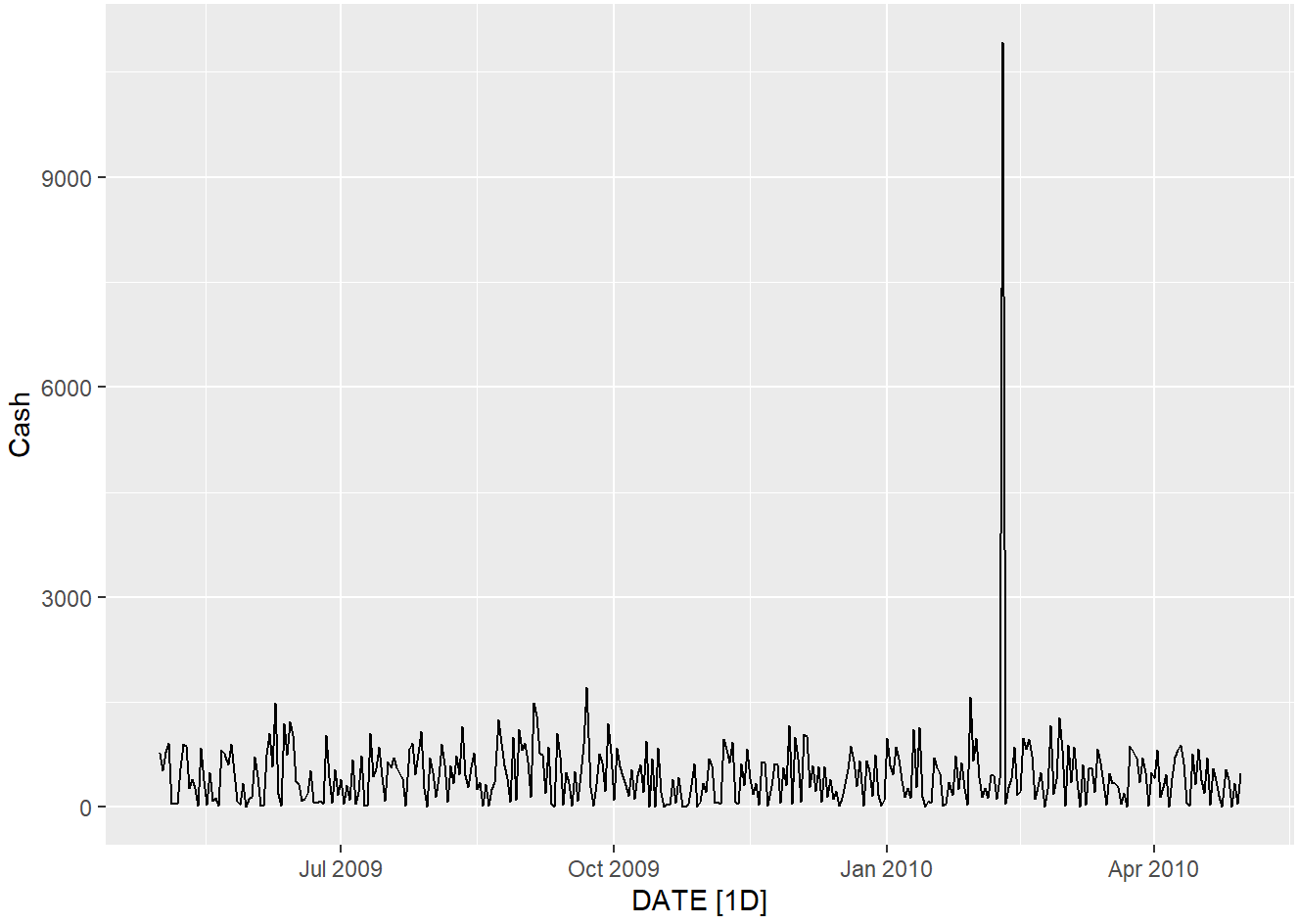
ATM 4 Work

```
#Chcking for zeros
print(atm4 |> filter(Cash==0)) # No Zeros
```

```
## # A tsibble: 0 x 4 [?]
## # Key:       ATM [0]
## # ℹ 4 variables: DATE <date>, ATM <chr>, Cash <dbl>, source <chr>
```

```
autoplot(atm4) # one large outlier.
```

```
## Plot variable not specified, automatically selected `.vars = Cash`
```



```
## Looking at ETS
auto_ets_atm4 <- atm4 |> model(auto = ETS(Cash), #ETS(M,N,A) is retrieved as the best automat
ic fit.
                               ANM = ETS(Cash ~ error("A") + trend("N") + season("M")),
                               MNM = ETS(Cash ~ error("M") + trend("N") + season("M")),
                               MNA = ETS(Cash ~ error("M") + trend("N") + season("A")))

print(auto_ets_atm4)
```

```
## # A mable: 1 x 5
## # Key:     ATM [1]
##   ATM           auto          ANM          MNM          MNA
##   <chr>      <model>      <model>      <model>      <model>
## 1 ATM4  <ETS(M,N,A)> <ETS(A,N,M)> <ETS(M,N,M)> <ETS(M,N,A)>
```

```
## ANA is best ETS fit.



print(auto_ets_atm4 |> report()) # Best model ETS(M,N,A)
```

```
## Warning in report.mdl_df(auto_ets_atm4): Model reporting is only supported for
## individual models, so a glance will be shown. To see the report for a specific
## model, use `select()` and `filter()` to identify a single model.
```

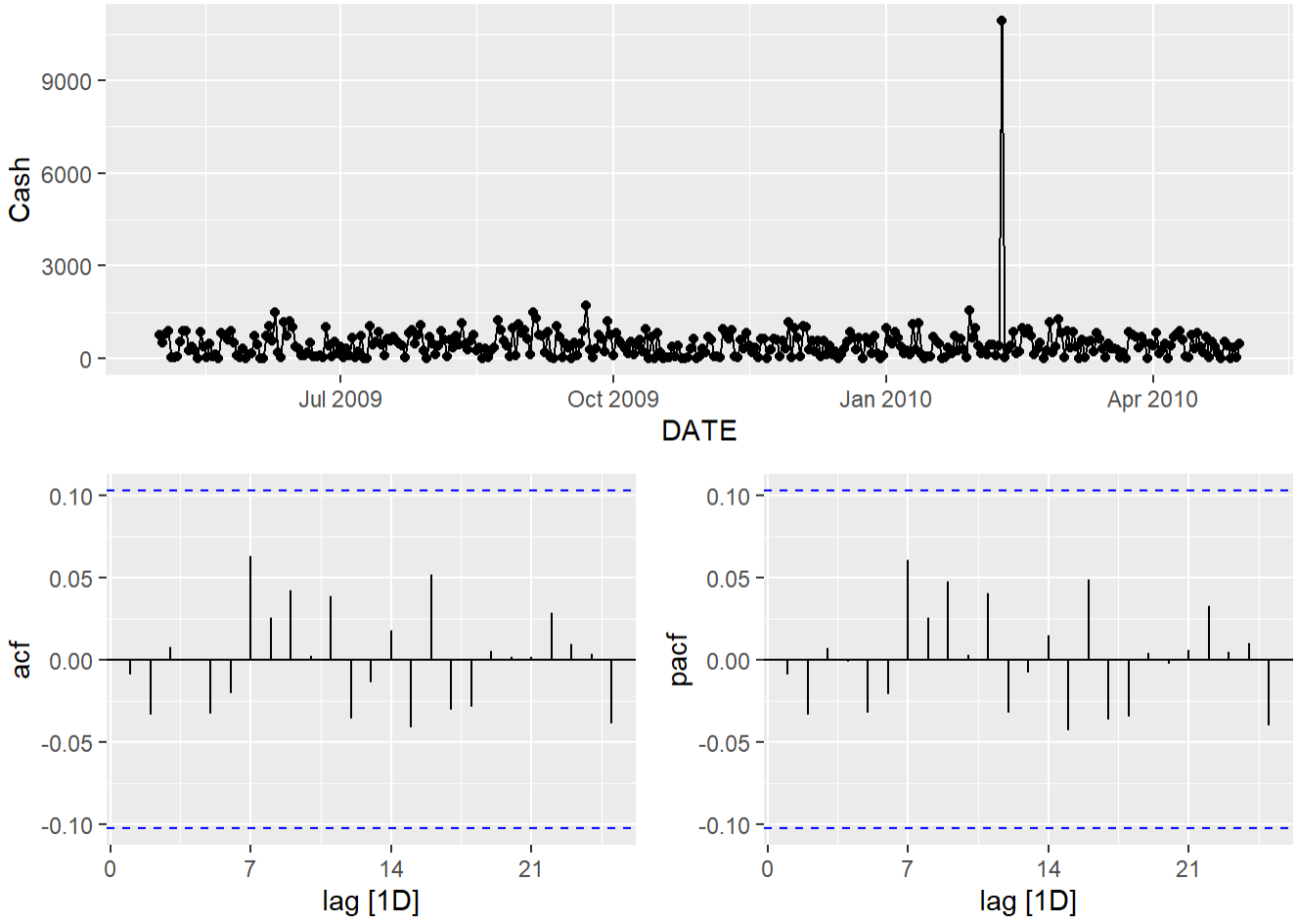
```
## # A tibble: 4 × 10
##   ATM   .model    sigma2 log_lik   AIC  AICc   BIC     MSE    AMSE     MAE
##   <chr> <chr>      <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl>
## 1 ATM4  auto        1.66  -3335. 6691. 6691. 6730. 416178. 416926.   0.777
## 2 ATM4  ANM    413836.    -3432. 6885. 6886. 6924. 403631. 404608. 301.
## 3 ATM4  MNM         1.85  -3388. 6795. 6796. 6834. 724334. 752241.   0.700
## 4 ATM4  MNA         1.66  -3335. 6691. 6691. 6730. 416178. 416926.   0.777
```

```
# AIC =6690.624, AICc= 6691.246, BIC=6729.623

print(auto_ets_atm4 |> accuracy())
```

```
## # A tibble: 4 × 11
##   ATM   .model .type       ME  RMSE   MAE   MPE  MAPE  MASE RMSSE     ACF1
##   <chr> <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1 ATM4  auto   Training  77.0  645.  312. -510.  552. 0.777 0.720 -0.0106
## 2 ATM4  ANM    Training  12.1  635.  301. -562.  592. 0.750 0.709 -0.00436
## 3 ATM4  MNM    Training -82.3  851.  377. -747.  779. 0.939 0.950 -0.00897
## 4 ATM4  MNA    Training  77.0  645.  312. -510.  552. 0.777 0.720 -0.0106
```

```
#RMSE 645.1182, not lowest but when using above indicotrs too Going with the auto selected mo
del.

## Rerunning with just the good models.
auto_ets_atm2 <- atm4 |> model(auto = ETS(Cash))


##   ------------------------------ Looking at ARIMA Models  ------------------------------

## Checking for Stationarity before modeling. No trend so may not need altering, or at least
a small amount of altering.

print(gg_tsdisplay(atm4,plot_type ="partial"))
```

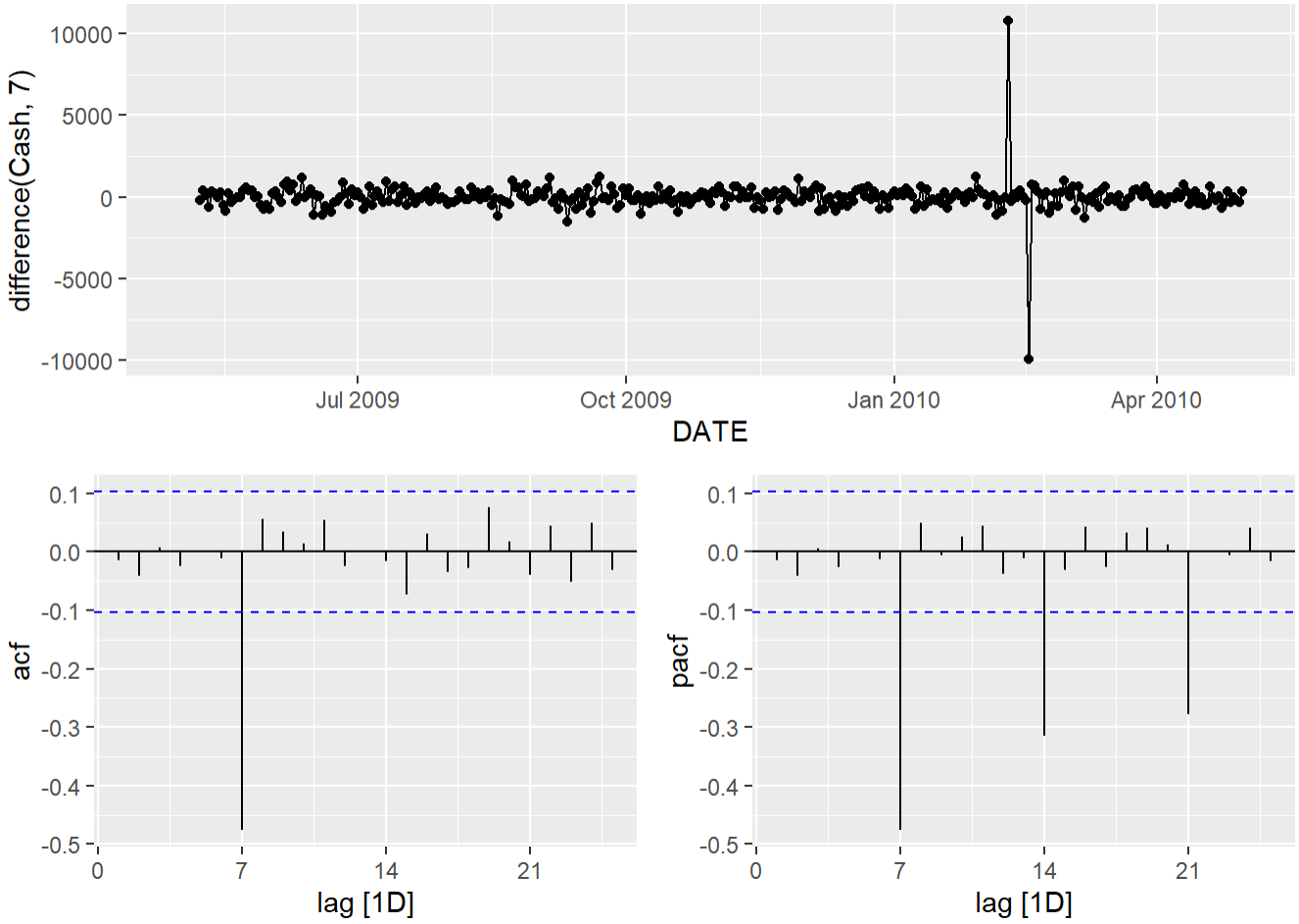
```
## Plot variable not specified, automatically selected `y = Cash`
```




```
# Mainly stationary already. No transformations may be needed, but will check anyway.


print(atm4 |> gg_tsdisplay(difference(Cash,7),plot_type ="partial"))
```

```
## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



```
# The 7 day seasonal difference looks much better, so doing that.
# ACF Shows some one outlier at 7
# PACF shows outliers  at 7 14 and 21. Biggest at 7. No non-seasonal outlier.


arima_atm4 <- atm4 |>
  model(manual_select2 = ARIMA(Cash ~ pdq(0,0,0) + PDQ(2,1,0)),
        manual_select3 = ARIMA(Cash ~ pdq(0,0,0) + PDQ(3,1,0)),
        auto_step = ARIMA(Cash), # SELECTED: <ARIMA(0,0,0) w/ mean>
        auto_search = ARIMA(Cash, stepwise = FALSE, approx=FALSE) )# ALSO SELECTED <<ARIMA
(0,0,0) w/ mean>>

## Auto Step and Auto search give same model: <ARIMA(0,0,0) w/ mean>
#print(arima_atm4)
# So best forecast is just the mean of the data.

print(arima_atm4 |> report())
```

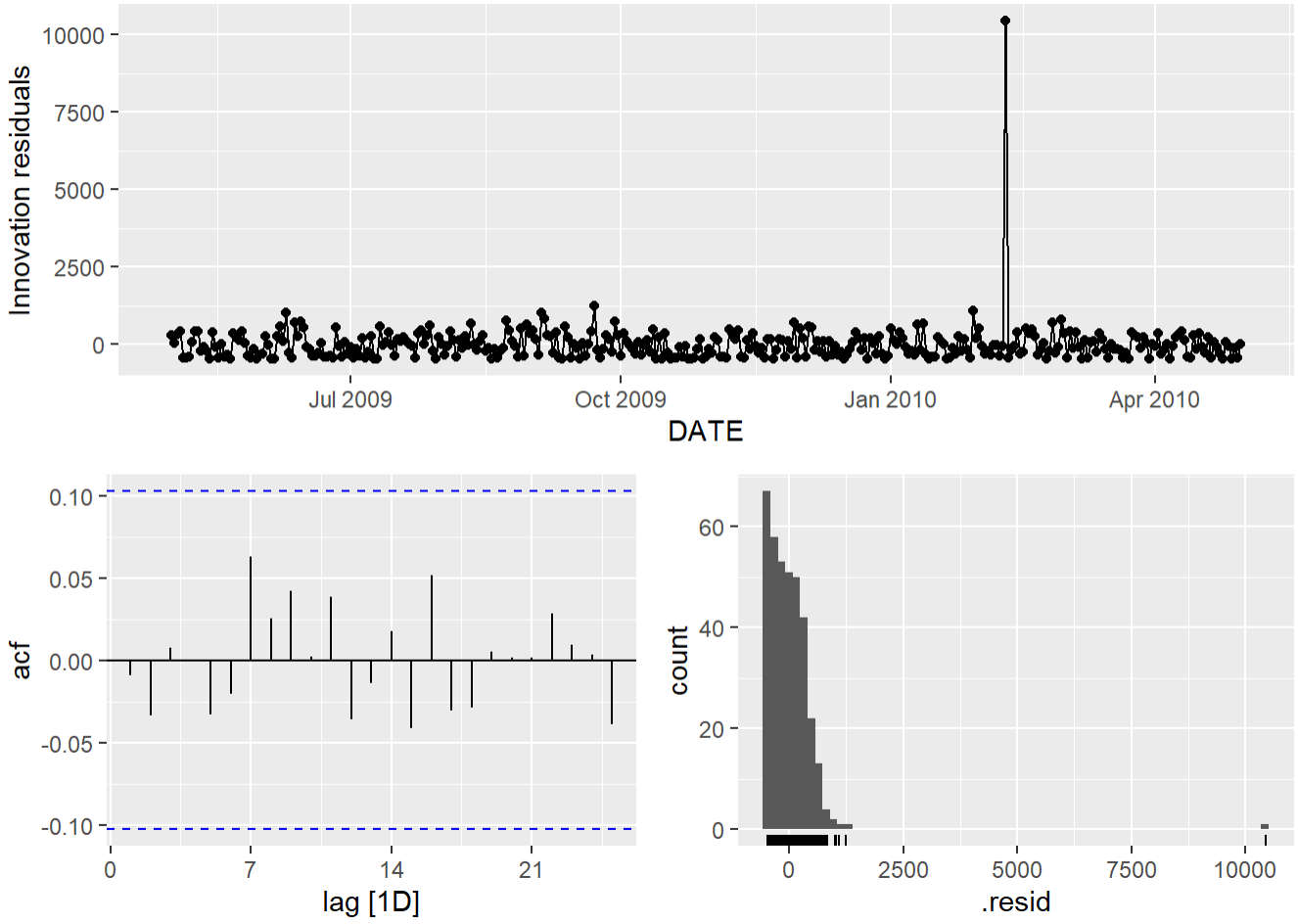
```
## Warning in report.mdl_df(arima_atm4): Model reporting is only supported for
## individual models, so a glance will be shown. To see the report for a specific
## model, use `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 4 × 9
##   ATM   .model          sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
##   <chr> <chr>            <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>
## 1 ATM4  manual_select2 562945.  -2879. 5763. 5763. 5775. <cpl [14]> <cpl [0]>
## 2 ATM4  manual_select3 518954.  -2864. 5737. 5737. 5752. <cpl [21]> <cpl [0]>
## 3 ATM4  auto_step      423718.  -2882. 5768. 5768. 5776. <cpl [0]>  <cpl [0]>
## 4 ATM4  auto_search    423718.  -2882. 5768. 5768. 5776. <cpl [0]>  <cpl [0]>
```

```
## Manually selected models are not as good as automated ones according to AIC, AICc and BIC.
I think auto models are the same, and better.
## <<ARIMA(0,0,0) w/ mean> AIC = 5768.064, AICc=5768.097, BIC=5775.864

print(arima_atm4 |> accuracy())
```

```
## # A tibble: 4 × 11
##   ATM   .model      .type        ME  RMSE   MAE   MPE  MAPE  MASE RMSSE     ACF1
##   <chr> <chr>       <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1 ATM4  manual_sel… Trai… -4.52e+ 0  741.  339. -547.  586. 0.844 0.827 -0.0110
## 2 ATM4  manual_sel… Trai… -4.54e+ 0  710.  329. -540.  577. 0.820 0.793 -0.00880
## 3 ATM4  auto_step   Trai… -1.51e-10  650.  324. -617.  647. 0.805 0.725 -0.00936
## 4 ATM4  auto_search Trai… -1.51e-10  650.  324. -617.  647. 0.805 0.725 -0.00936
```

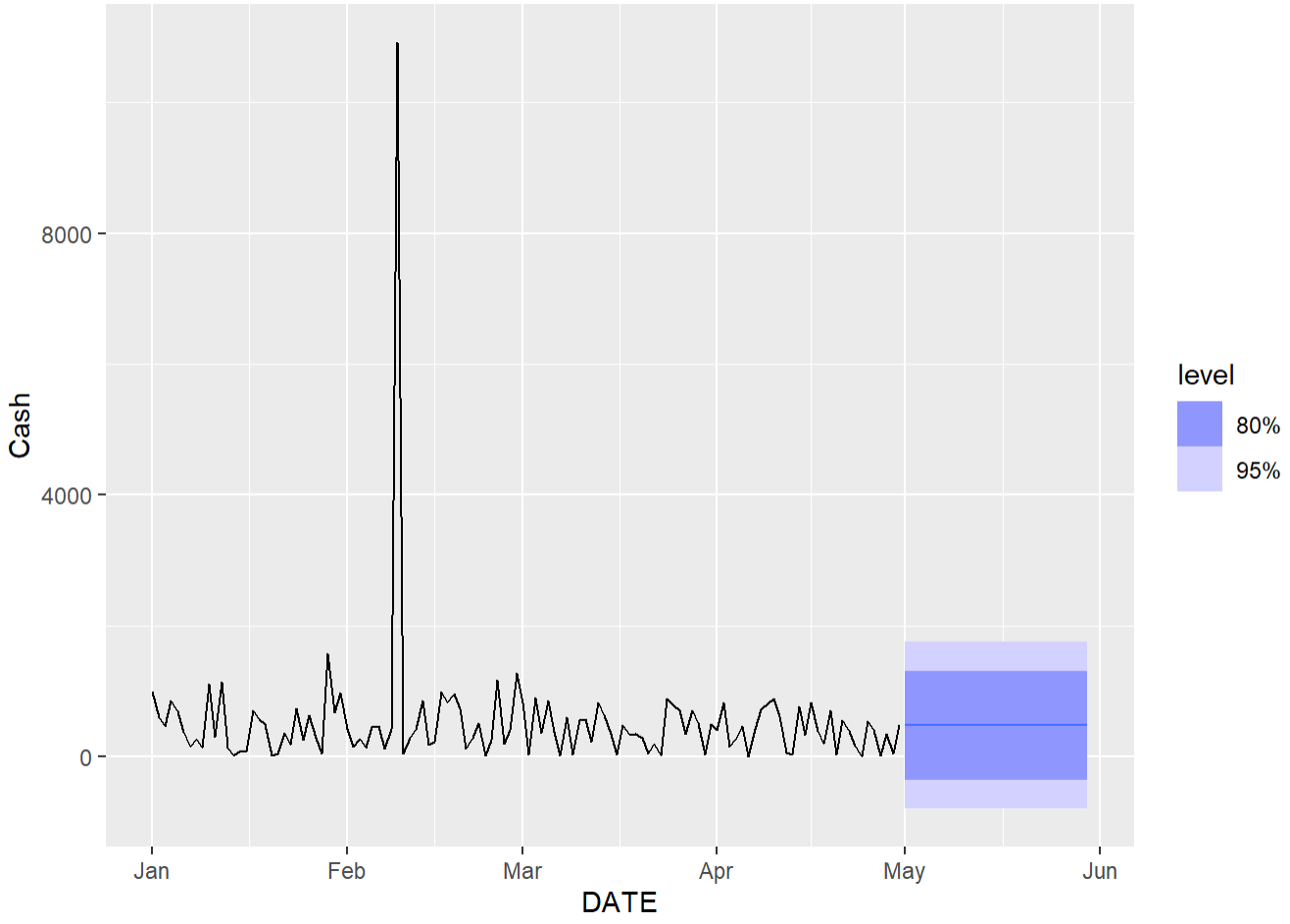
```
## Confirming that auto_step is the best model. RMSE is 650.0437


## ---- COMPARING ETS AND ARIMA ----
### AUTOSTEP ARIMA -> RMSE = 650.0437, AIC = 5768.064, AICc=5768.097, BIC=5775.864
### MNA ETS -> RMSE=645.1182, AIC =6690.624, AICc= 6691.246, BIC=6729.623


## OVERALL the ARIMA numbers are much better. Will be using ARIMA to forecast.

## Running again with only selected model
arima_atm4 <- atm4 |>
  model(auto_step = ARIMA(Cash), # SELECTED: ARIMA(0,0,0) w/ mean>>
        )


## Last levels of confirmation checks
## looking at residuals
print(gg_tsresiduals(arima_atm4))
```



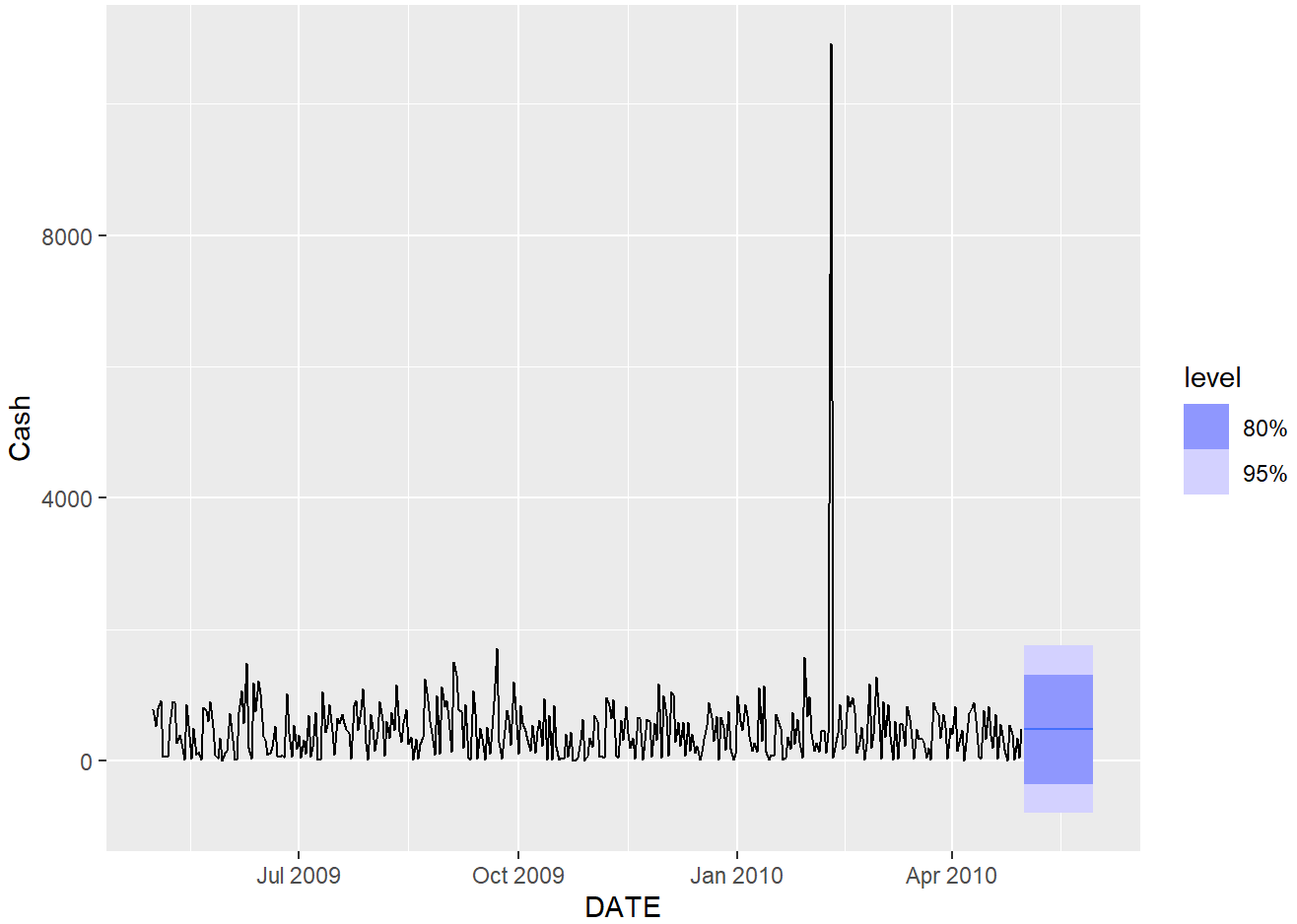

```
# Ljung Box test
print(augment(arima_atm4) |> features(.innov, ljung_box, lag=7, dof=0)) # pval .92
```

```
## # A tibble: 1 × 4
##   ATM   .model    lb_stat lb_pvalue
##   <chr> <chr>       <dbl>     <dbl>
## 1 ATM4  auto_step    2.51     0.926
```

```
# NO autocorrelation left in the residuals so its good. Moving forward with forecast


### With model selected taking a look at the residuals for the
atm4_forecast <- arima_atm4 |>
  forecast(h = 30) #30 days

# ATM 4 forecast
print(atm4_forecast |>
    autoplot(atm4|>filter(DATE>=as.Date('2010-01-01'))))
```




```
#p <- atm4_forecast |> autoplot(atm4|>filter(DATE>=as.Date('2010-01-01')))
#ggsave("images/atm4_proj_lim.png", plot = p, width = 12, height = 8, dpi = 300)

print(atm4_forecast |>
    autoplot(atm4))
```




```
#p <- atm4_forecast |> autoplot(atm4)
#ggsave("images/atm4_proj.png", plot = p, width = 12, height = 8, dpi = 300)


## Looking at forecasted values
#atm4_forecast|> hilo() |> as_tsibble()

#forecast_table <- atm4_forecast |> hilo() |> as_tsibble()

# Write to CSV
#write.csv(atm4_forecast, "projection_data/atm4_forecast_values.csv", row.names = FALSE)
```

APPENDIX B - Part B Code & Analysis

```
# RAW FILE ALSO SITS HERE: https://raw.githubusercontent.com/jhnboyy/CUNY_SPS_WORK/main/Spring2025/DATA624/Project1/ResidentialCustomerForecastLoad-624.xlsx
res_df<-read_excel('ResidentialCustomerForecastLoad-624.xlsx')
#res_df
```

```
summary(res_df)
```

```
##   CaseSequence     YYYY-MMM              KWH         
##  Min.   :733.0   Length:192         Min.   :  770523  
##  1st Qu.:780.8   Class :character   1st Qu.: 5429912  
##  Median :828.5   Mode  :character   Median : 6283324  
##  Mean   :828.5                      Mean   : 6502475  
##  3rd Qu.:876.2                      3rd Qu.: 7620524  
##  Max.   :924.0                      Max.   :10655730  
##                                     NA's   :1
```

```
## There is one null in KWH
## No Nulls in CaseSequence

res_df |> filter(is.na('YYYY-MMM'))
```

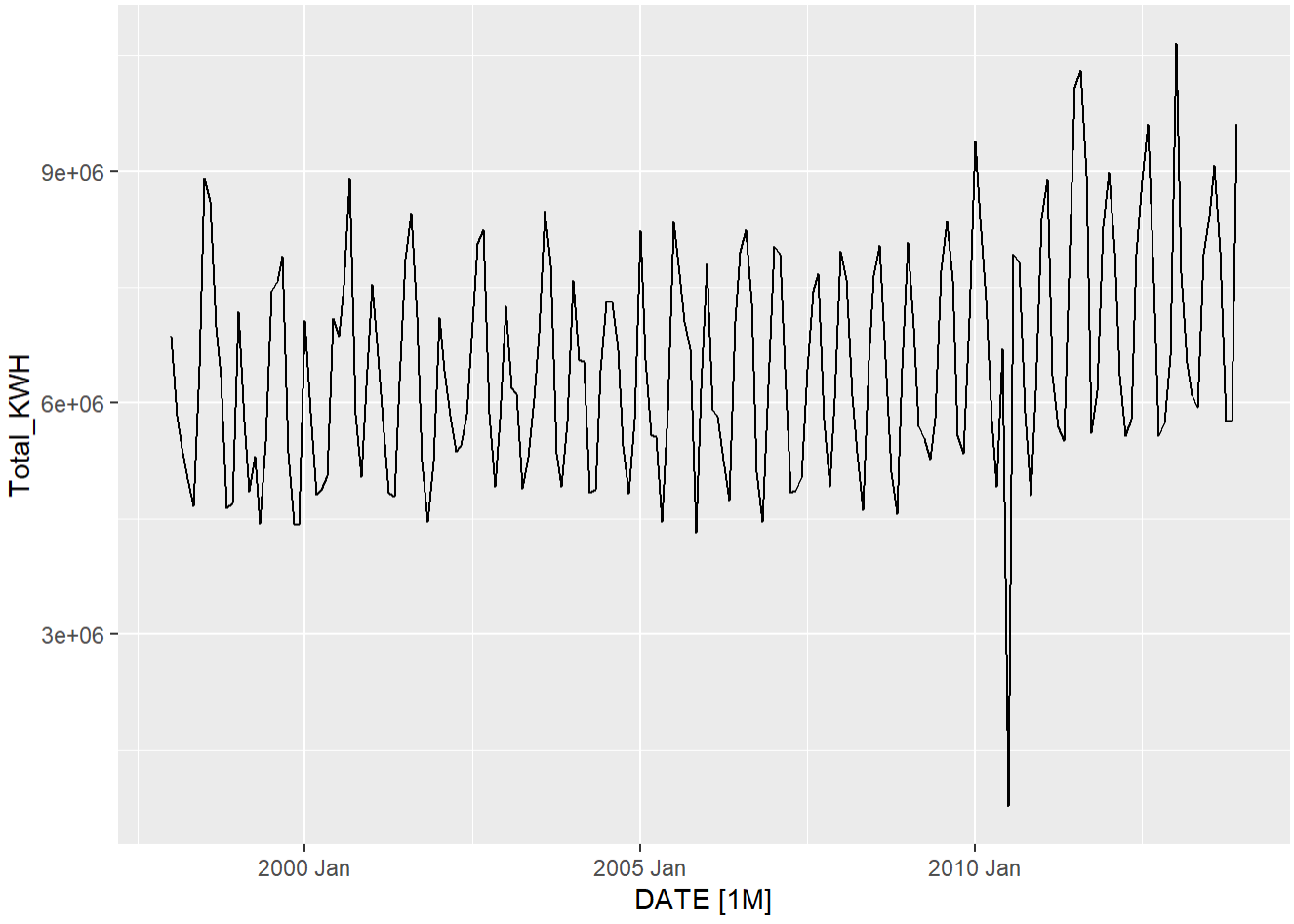
```
## # A tibble: 0 × 3
## # ℹ 3 variables: CaseSequence <dbl>, YYYY-MMM <chr>, KWH <dbl>
```

```
## No null in dates

## Looking at the one null
res_df|> filter(is.na(KWH))
```

```
## # A tibble: 1 × 3
##   CaseSequence `YYYY-MMM`   KWH
##          <dbl> <chr>      <dbl>
## 1          861 2008-Sep      NA
```

```
## The one null in KWH is in September 2008, checking the before and after

res_df<- res_df |>
    mutate(DATE = yearmonth(`YYYY-MMM`),
           source = if_else(is.na(KWH), "imputed", "original"))

#res_df |> filter( DATE >= yearmonth('2008-Aug') & DATE<=yearmonth('2008-Oct'))

red_df_imp <- res_df |> mutate(KWH = if_else(is.na(KWH) & DATE >= yearmonth('2008-Aug') & DAT
E<=yearmonth('2008-Oct'),
                                    (lag(KWH) + lead(KWH)) / 2, KWH ))
## Checking imp
#red_df_imp |> filter(DATE>=yearmonth('2008-Aug'),DATE<=yearmonth('2008-Oct'))

## Placing into a Tsible with the Case Sequence removed
red_tsibble <- red_df_imp |>
  as_tsibble(index = 'DATE', key = CaseSequence)|>
  summarize(Total_KWH = sum(KWH))

# plotting
print(red_tsibble|> autoplot())
```

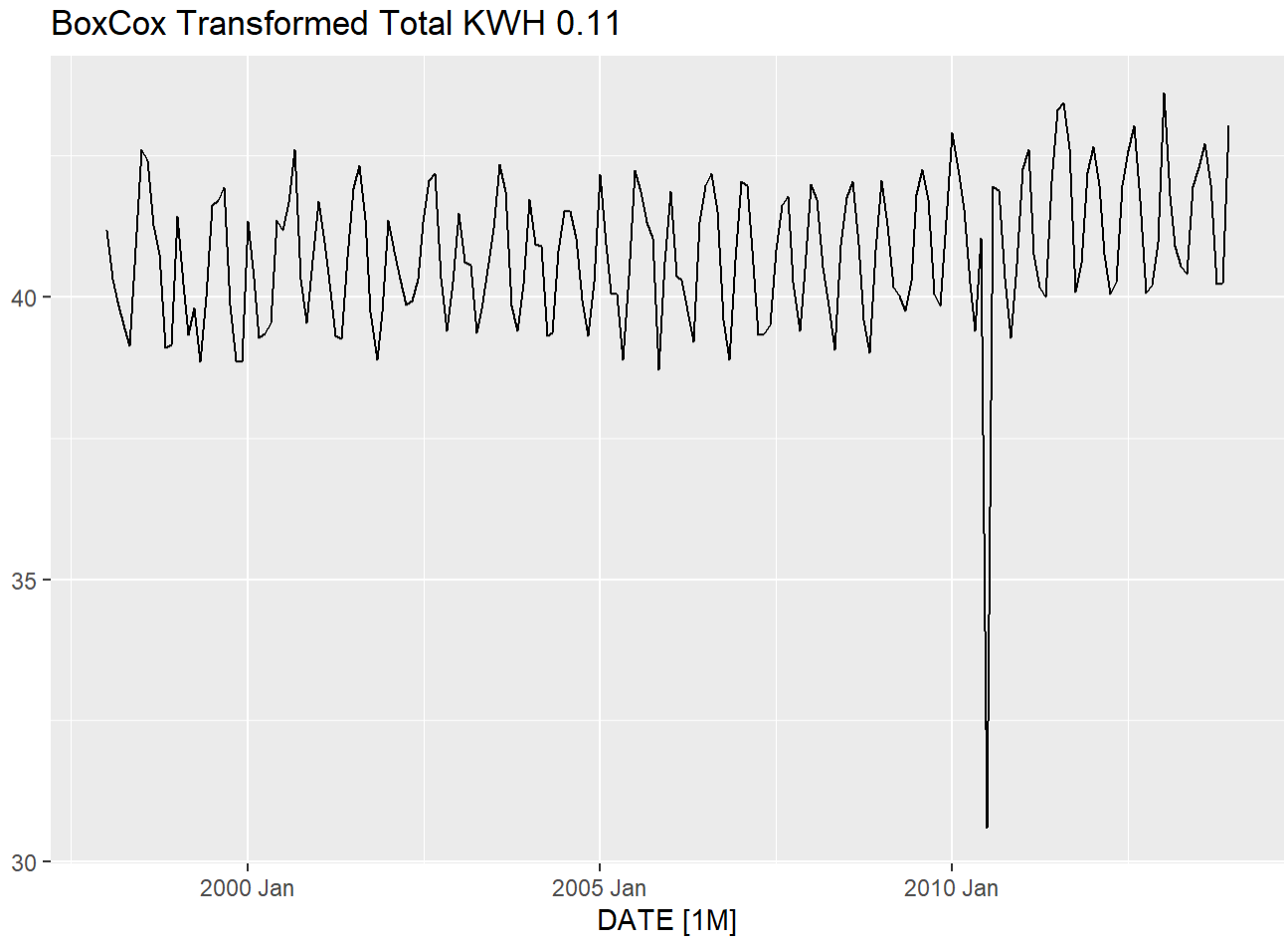
```
## Plot variable not specified, automatically selected `.vars = Total_KWH`
```



```
#p <- red_tsibble|> autoplot()
#ggsave("images/partb_dataplot.png", plot = p, width = 12, height = 8, dpi = 300)
## Very subtle trend with seasonal variation. One large drop / outlier.

## Using Box -Cox for best transformation
lambda <- red_tsibble |>
  features(Total_KWH, features = guerrero) |>
  pull(lambda_guerrero)
print(lambda) ## 0.107 Close to a log transformation lambda
```

```
## [1] 0.1073943
```

```
## Transformed data looks good
print(red_tsibble |>  autoplot(box_cox(Total_KWH, lambda)) +  labs(y = "",title = latex2ex
p::TeX(paste0("BoxCox Transformed Total KWH ",round(lambda,2)))))
```



```
## Jumping right into ARIMA models because ARIMA was the better model for nearly all of part 1.
## Similiarly only doing the automated best model find, as in part 1 nearly all the time the function was correct.


red_model <- red_tsibble |>
  model(auto_step = ARIMA(box_cox(Total_KWH, lambda)), # SELECTED: <ARIMA(0,1,2)(0,0,2)[12]>
        auto_search = ARIMA(box_cox(Total_KWH, lambda), stepwise = FALSE, approx=FALSE) )# Selected:<ARIMA(0,1,2)(2,0,0)[12]>
## Seeing Selections
print(red_model)
```

```
## # A mable: 1 x 2
##                    auto_step                auto_search
##                      <model>                    <model>
## 1 <ARIMA(0,1,2)(0,0,2)[12]> <ARIMA(0,1,2)(2,0,0)[12]>
```

```
## Selected to slightly different models, now going to compare both for best one.
print(red_model|> report())
```

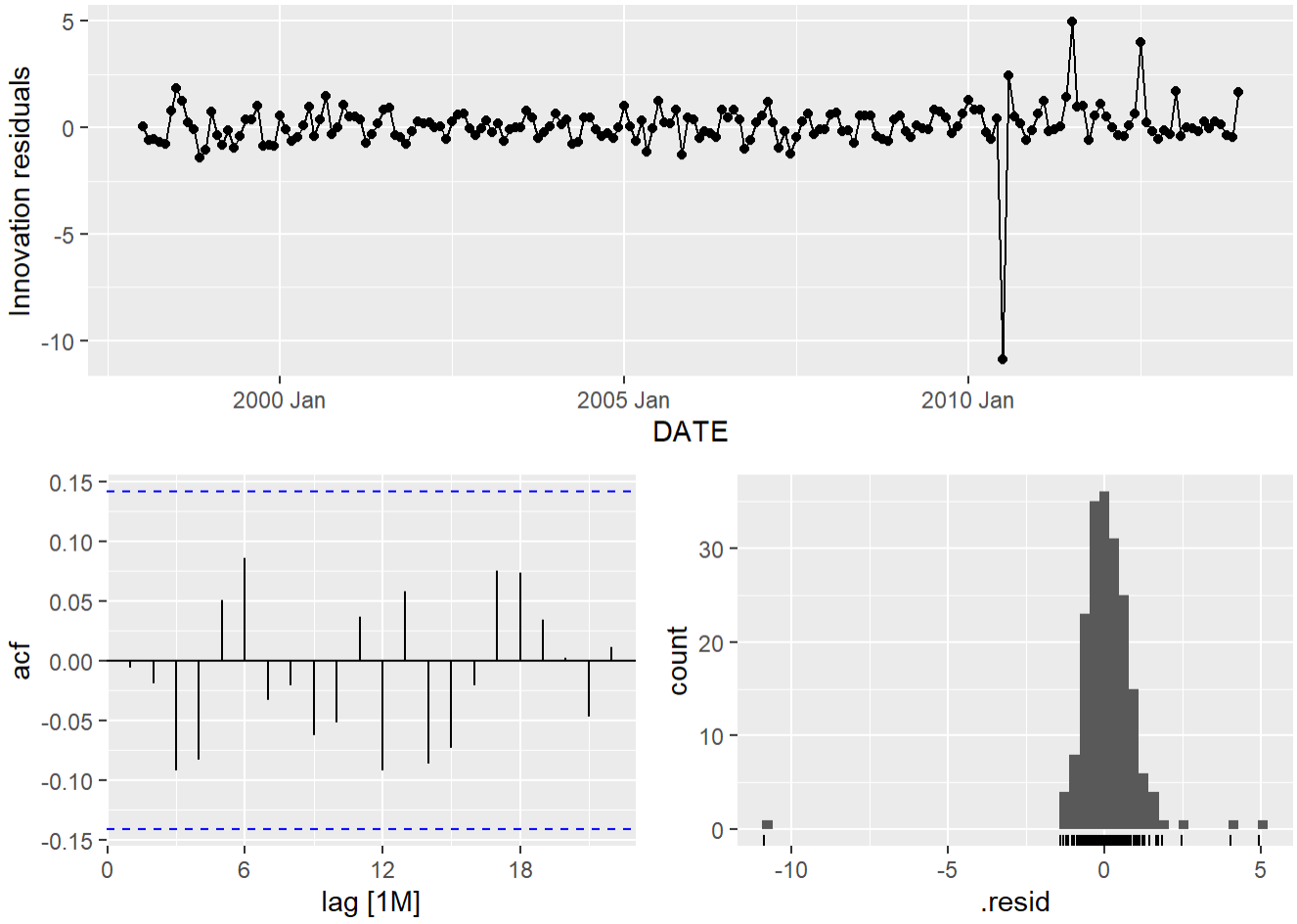
```
## Warning in report.mdl_df(red_model): Model reporting is only supported for
## individual models, so a glance will be shown. To see the report for a specific
## model, use `select()` and `filter()` to identify a single model.
```

```
## # A tibble: 2 × 8
##   .model      sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
##   <chr>        <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>
## 1 auto_step     1.43   -305.  620.  620.  636. <cpl [0]>  <cpl [26]>
## 2 auto_search   1.28   -296.  601.  602.  618. <cpl [24]> <cpl [2]>
```

```
#auto_search (<ARIMA(0,1,2)(2,0,0)[12]>) is better: AIC:601.4199,   AICc:601.7443,  BIC:617.6813
print(red_model|> accuracy())
```

```
## # A tibble: 2 × 10
##   .model      .type         ME     RMSE     MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>       <chr>      <dbl>    <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 auto_step   Training 232730. 1197100. 855264. -2.97  16.4  1.23 1.02  0.137
## 2 auto_search Training 205658. 1088172. 700774. -2.90  14.1  1.01 0.924 0.111
```

```
## for error measures the auto_search is also beterr with lower error values accross the tabl
e.

## Redoing the model with just the better one in preparation for forecasting.
red_model <- red_tsibble |>
  model(auto = ARIMA(box_cox(Total_KWH, lambda), stepwise = FALSE, approx=FALSE) )# Selecte
d:<ARIMA(0,1,2)(2,0,0)[12]>

## Last levels of confirmation checks
## looking at residuals
print(gg_tsresiduals(red_model)) # Fine with an outlier or 2
```



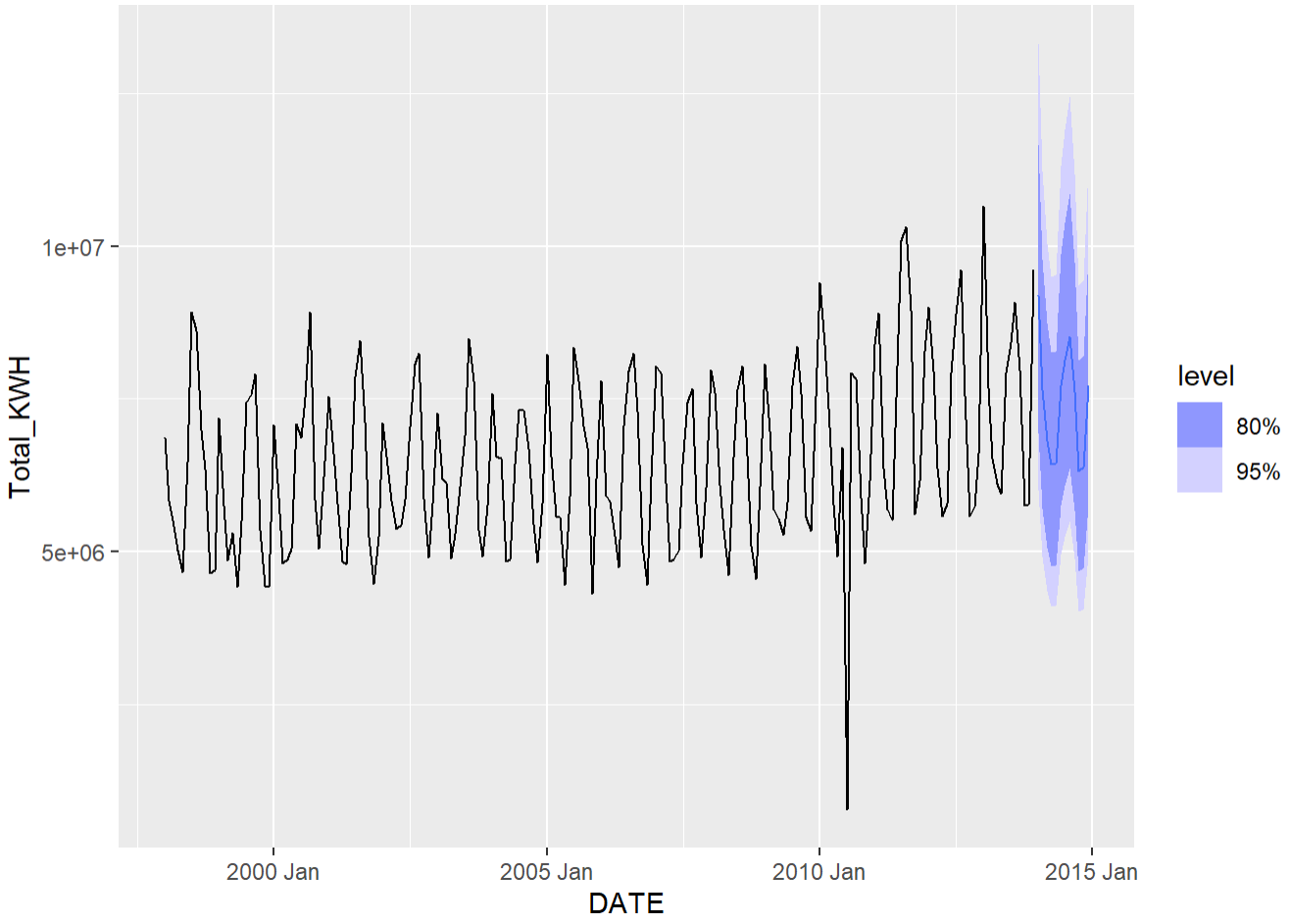

```
# Ljung Box test
print(augment(red_model) |> features(.innov, ljung_box, lag=12, dof=4)) # pval .35
```

```
## # A tibble: 1 × 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 auto      8.79     0.360
```

```
print(augment(red_model) |> features(.innov, ljung_box, lag=24, dof=4)) # pval .45
```

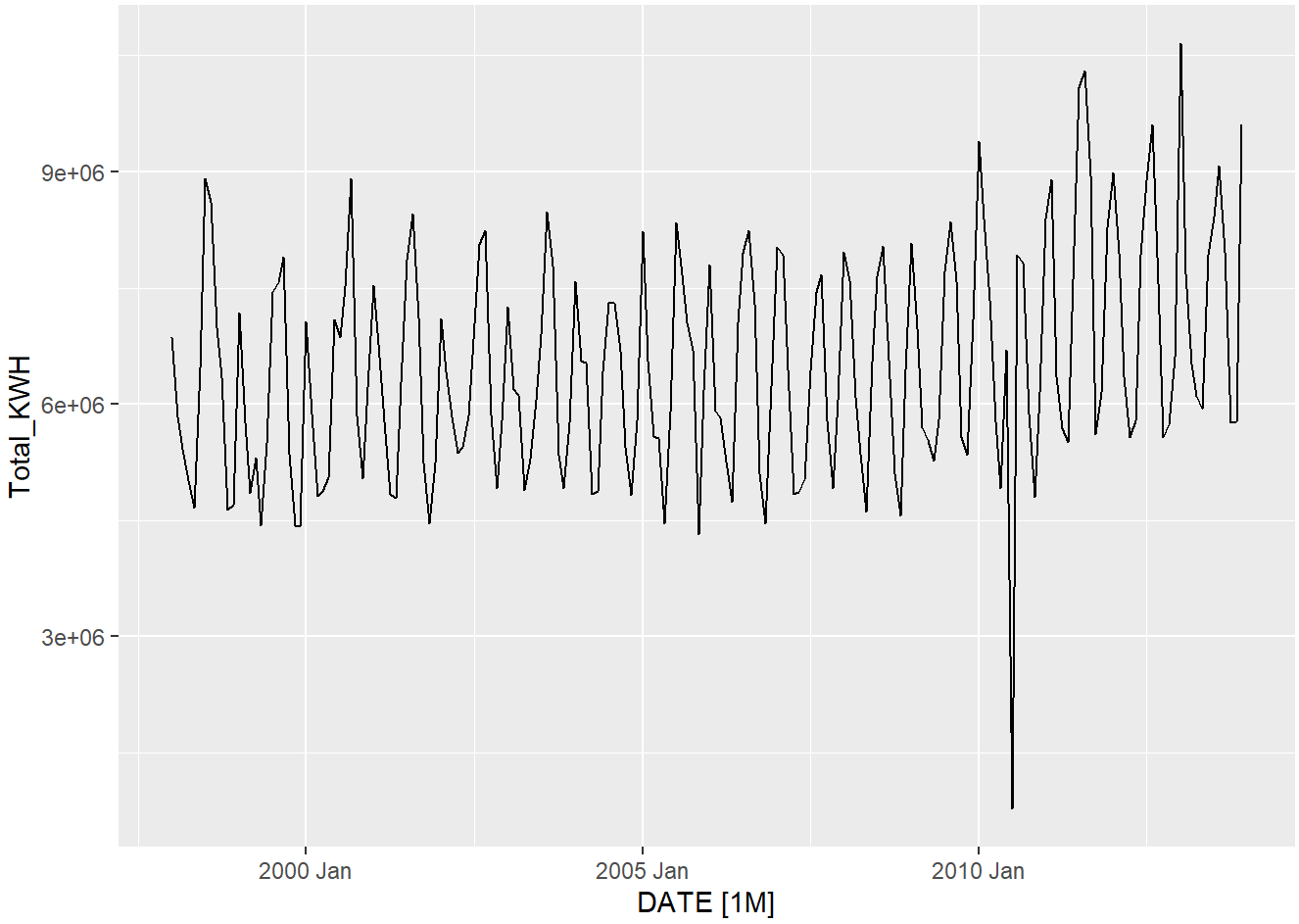
```
## # A tibble: 1 × 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 auto      20.0     0.458
```

```
# NO autocorrelation left in the residuals so its good. Moving forward with forecast


### With model selected taking a look at the residuals for the
red_forecast <- red_model |> forecast(h = 12) #12 months

# KWH forecast
print(red_forecast |>
    autoplot(red_tsibble))
```




```
#p <- red_forecast |>
    autoplot(red_tsibble)
```

```
## Plot variable not specified, automatically selected `.vars = Total_KWH`
```




```
#ggsave("images/partb_data_forecast.png", plot = p, width = 12, height = 8, dpi = 300)

## Looking at forecasted values
#red_forecast|> hilo() |> as_tsibble()

#forecast_table <- red_forecast |> hilo() |> as_tsibble()

#write.csv(red_forecast, "projection_data/red_forecast_values.csv", row.names = FALSE)
```