Project 1

Victor Torres

2024-10-25

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 Apr 11. I will accept late submissions with a penalty until the meetup after that when we review some projects.

Parts

Libraries

```
library(readxl)
library(fpp3)
```

```
## Warning: package 'fpp3' was built under R version 4.4.1
```

```
## Registered S3 method overwritten by 'tsibble':
##   method               from
##   as_tibble.grouped_df dplyr
```

```
## -- Attaching packages --------------------------------------------- fpp3 1.0.0 --
```

```
## v tibble      3.2.1     v tsibble     1.1.5
## v dplyr       1.1.4     v tsibbledata 0.4.1
## v tidyr       1.3.1     v feasts      0.3.2
## v lubridate   1.9.3     v fable       0.3.4
## v ggplot2     3.5.1     v fabletools  0.4.2
```

```
## Warning: package 'tsibble' was built under R version 4.4.1
```

```
## Warning: package 'tsibbledata' was built under R version 4.4.1
```

```
## Warning: package 'feasts' was built under R version 4.4.1
```

```
## Warning: package 'fabletools' was built under R version 4.4.1
```

```
## Warning: package 'fable' was built under R version 4.4.1
```

```
## -- Conflicts ------------------------------------------------- fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats 1.0.0     v readr   2.1.5
## v purrr   1.0.2     v stringr 1.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()     masks stats::filter()
## x tsibble::interval() masks lubridate::interval()
## x dplyr::lag()        masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.4.1

##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.4.1

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

```
library(dplyr)
library(lubridate)
library(tsibble)
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.4.1
```

```
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.4.1
```

```
##
## Attaching package: 'zoo'
##
## The following object is masked from 'package:tsibble':
##
##     index
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(xlsx)
```

```
## Warning: package 'xlsx' was built under R version 4.4.1
```

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.

```
# Read Excel file into R
ATM_Data <- read_excel("C:/Users/vitug/OneDrive/Desktop/CUNY Masters/DATA_624/ATM624Data.xlsx")
head(ATM_Data)
```

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

```
str(ATM_Data)
```

```
## tibble [1,474 x 3] (S3: tbl_df/tbl/data.frame)
##  $ DATE: num [1:1474] 39934 39934 39935 39935 39936 ...
##  $ ATM : chr [1:1474] "ATM1" "ATM2" "ATM1" "ATM2" ...
##  $ Cash: num [1:1474] 96 107 82 89 85 90 90 55 99 79 ...
```

```
# Converting DATE column format from numeric to date (YYYY-MM-DD).
ATM_Data$DATE <- as.Date(ATM_Data$DATE, origin = "1899-12-30")
# convert to tsibble
ATM_Data <-as_tsibble(ATM_Data, key = ATM, index = DATE)
# Summary of the converted data
summary(ATM_Data)
```

```
##       DATE                ATM                Cash          
##  Min.   :2009-05-01   Length:1474        Min.   :    0.0  
##  1st Qu.:2009-08-01   Class :character   1st Qu.:    0.5  
##  Median :2009-11-01   Mode  :character   Median :   73.0  
##  Mean   :2009-10-31                      Mean   :  155.6  
##  3rd Qu.:2010-02-01                      3rd Qu.:  114.0  
##  Max.   :2010-05-14                      Max.   :10919.8  
##                                          NA's   :19
```

```
# check NA values in columns
ATM_Data %>%
  filter(is.na(Cash))
```

```
## # A tsibble: 19 x 3 [1D]
## # Key:       ATM [3]
##    DATE       ATM    Cash
##    <date>     <chr> <dbl>
##  1 2009-06-13 ATM1     NA
##  2 2009-06-16 ATM1     NA
##  3 2009-06-22 ATM1     NA
##  4 2009-06-18 ATM2     NA
##  5 2009-06-24 ATM2     NA
##  6 2010-05-01 <NA>     NA
##  7 2010-05-02 <NA>     NA
##  8 2010-05-03 <NA>     NA
##  9 2010-05-04 <NA>     NA
## 10 2010-05-05 <NA>     NA
## 11 2010-05-06 <NA>     NA
## 12 2010-05-07 <NA>     NA
## 13 2010-05-08 <NA>     NA
## 14 2010-05-09 <NA>     NA
## 15 2010-05-10 <NA>     NA
## 16 2010-05-11 <NA>     NA
## 17 2010-05-12 <NA>     NA
## 18 2010-05-13 <NA>     NA
## 19 2010-05-14 <NA>     NA
```

```
#remove data with blank ATM information
ATM_Data <- ATM_Data[!is.na(ATM_Data$ATM),]

#view remaining missing data
ATM_Data %>%
  filter(is.na(Cash))
```

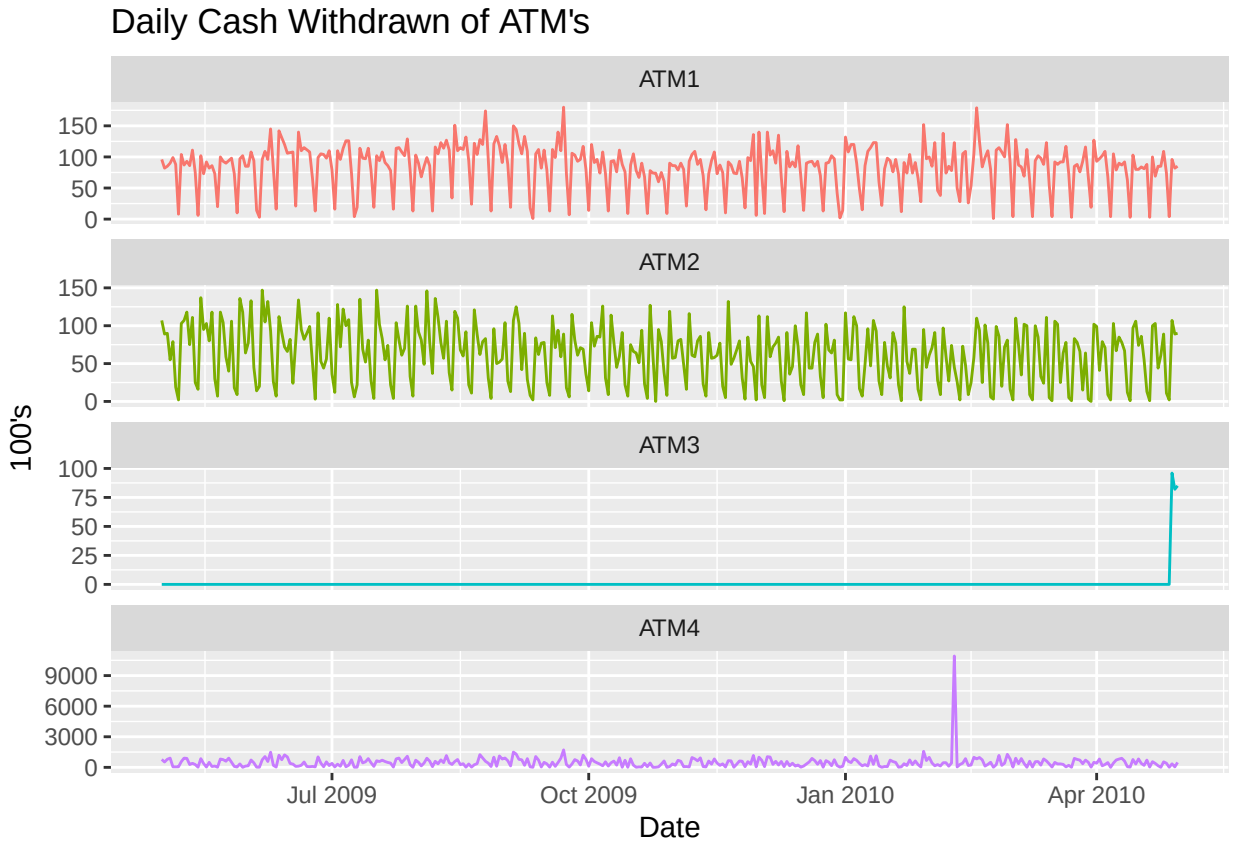
```
## # A tsibble: 5 x 3 [1D]
## # Key:       ATM [2]
##   DATE       ATM    Cash
##   <date>     <chr> <dbl>
## 1 2009-06-13 ATM1     NA
## 2 2009-06-16 ATM1     NA
## 3 2009-06-22 ATM1     NA
## 4 2009-06-18 ATM2     NA
## 5 2009-06-24 ATM2     NA
```

```r
# use na.approx to fill missing values
ATM_Data <- ATM_Data%>%
  mutate(Cash = na.approx(Cash))

#view the rows to confirm missing values
ATM_Data[c(44,47,49,53,55),]
```

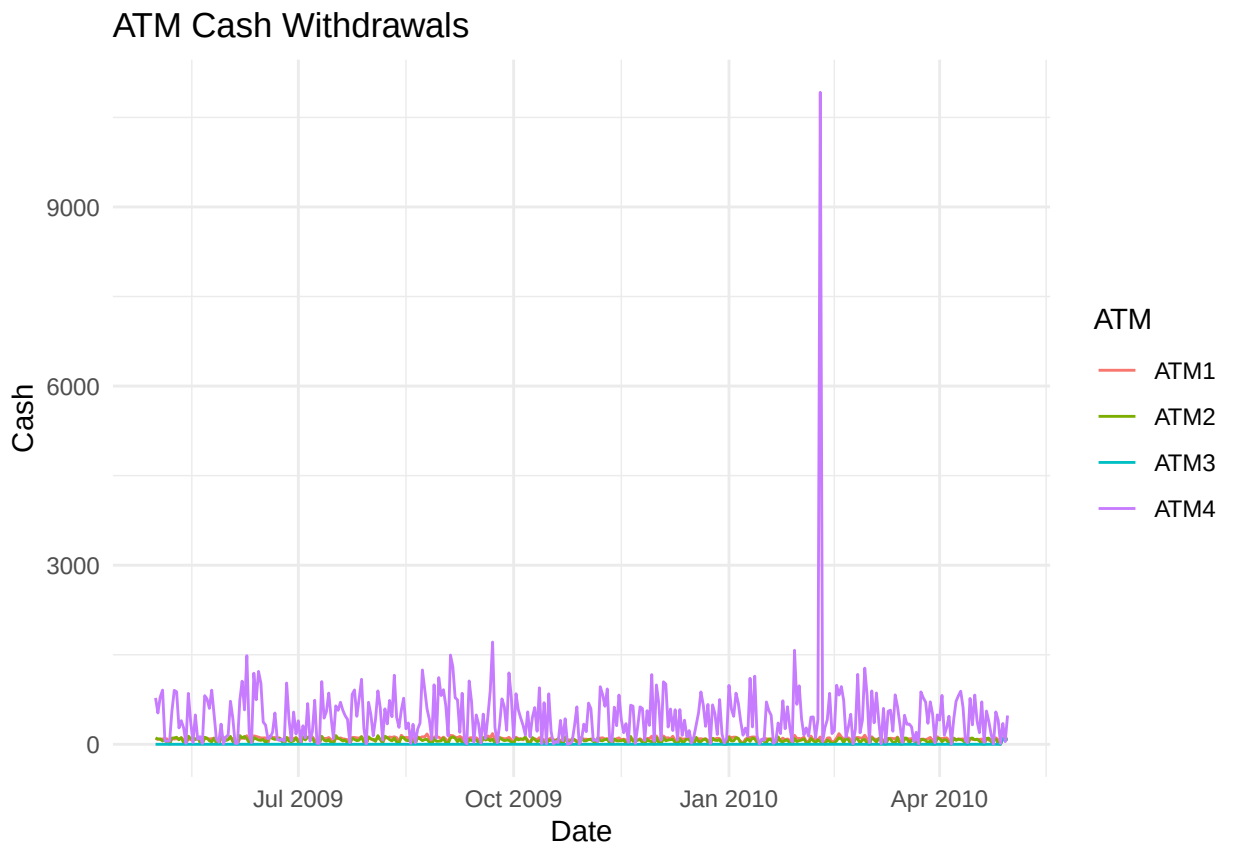
```
## # A tsibble: 5 x 3 [1D]
## # Key:       ATM [1]
##   DATE       ATM    Cash
##   <date>     <chr> <dbl>
## 1 2009-06-13 ATM1   131
## 2 2009-06-16 ATM1   107
## 3 2009-06-18 ATM1    21
## 4 2009-06-22 ATM1   112.
## 5 2009-06-24 ATM1    66
```

Time Series ATM data seems to have duplicate values when I used the summarize function to find missing values on it. The data seems to have seasonality which is not consistent. I decided to used a STL decomposition based on the data s behavior. The next step was to split up the data into training and testing for the forecast. I decided to apply ETS model to determine if the data has white noise before running the forecast. Also, I applied the ARIMA model to check if the data is stationary or not,after running several models, the ETS model has the lowest RMSE, which it might be the best option for this dataset.

```r
#time series plot
ATM_Data %>% ggplot(aes(x = DATE, y = Cash, col = ATM)) +
    geom_line(show.legend = FALSE) +
    facet_wrap(~ ATM, ncol = 1, scales = "free_y") +
    labs(title = "Daily Cash Withdrawn of ATM's ", x = "Date") +
    scale_y_continuous("100's")
```

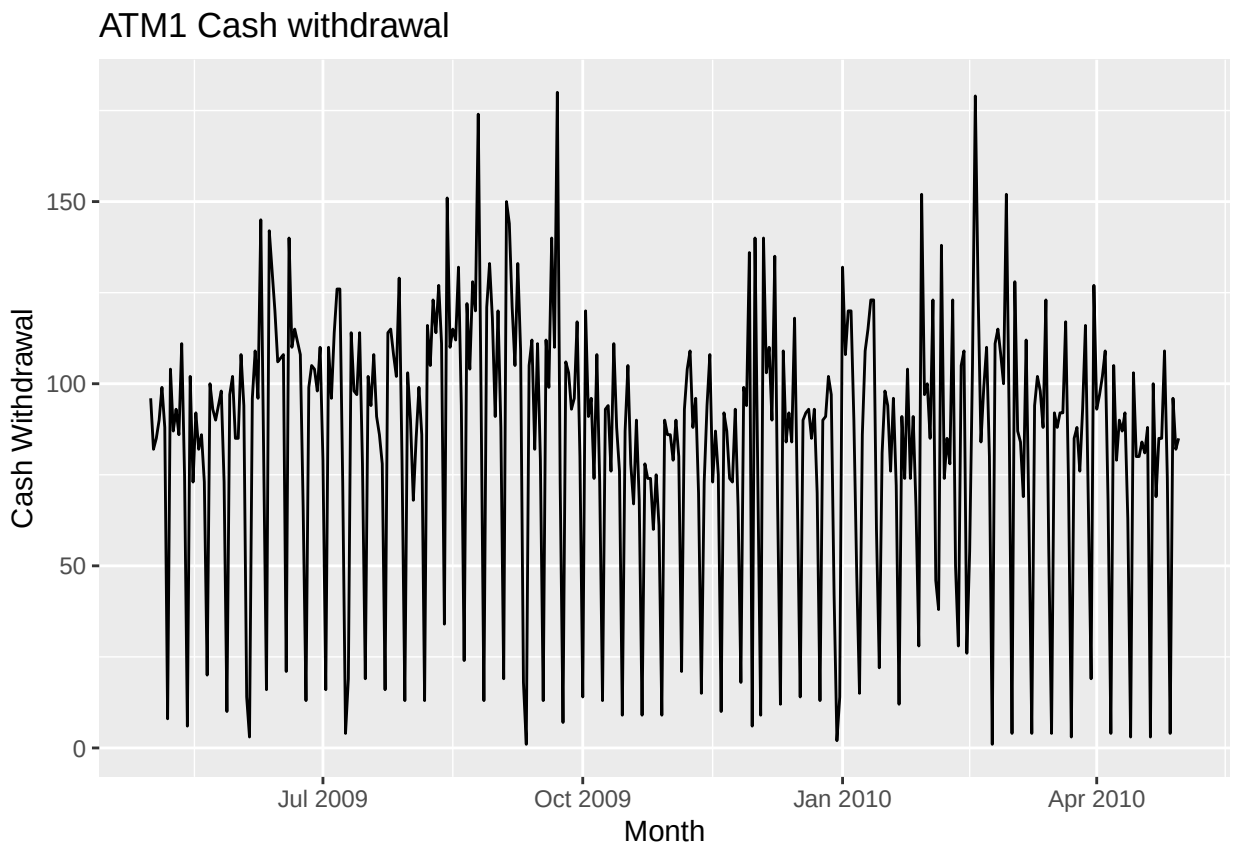



```
# Time Series Graph by ATM
ggplot(ATM_Data, aes(x = DATE, y = Cash, color = ATM)) +
  geom_line() +
  labs(title = "ATM Cash Withdrawals", x = "Date", y = "Cash") +
  theme_minimal()
```



ATM Models

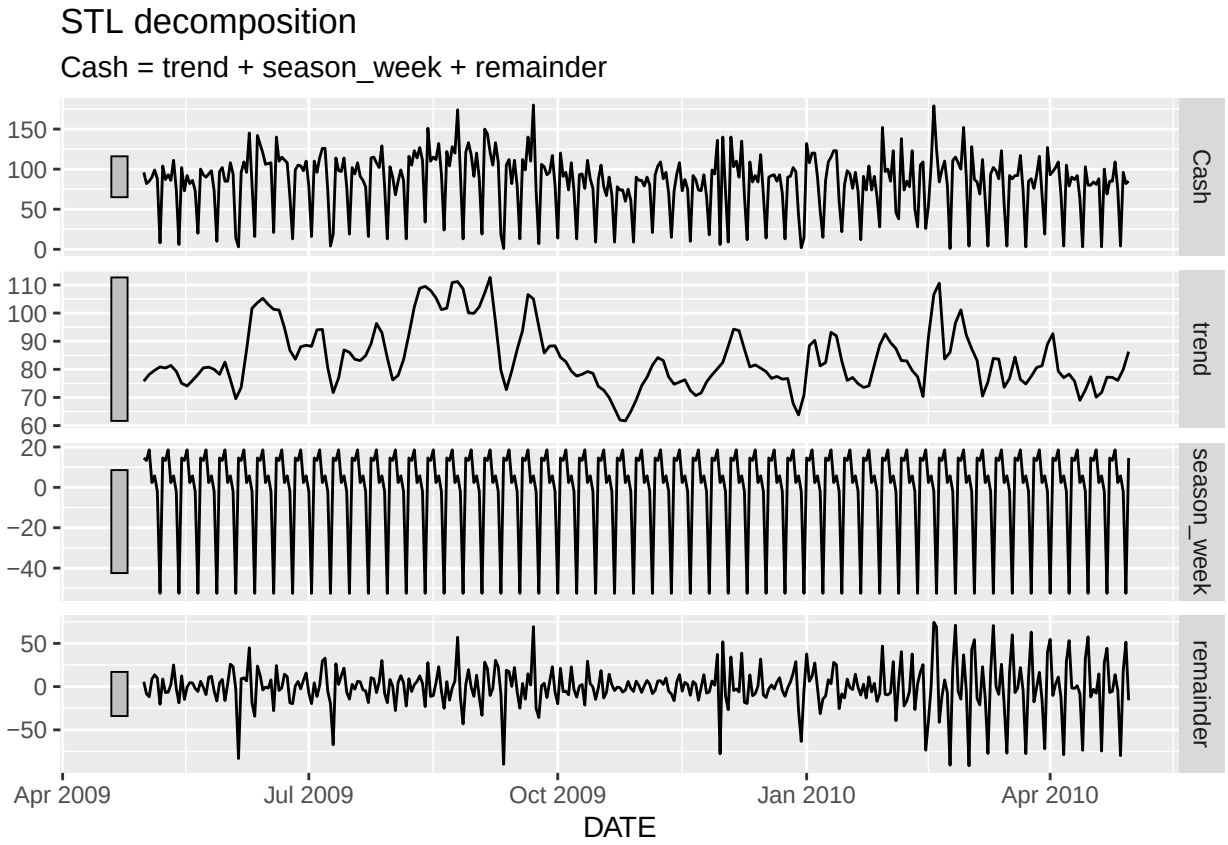
```
# ATM 1
ATM_Data %>%
  filter(ATM == 'ATM1') %>%
  autoplot(Cash) +
  labs(title = "ATM1 Cash withdrawal",x = "Month", y = "Cash Withdrawal")
```



ATM1

```
# Filter for ATM1 data and sum up the totals in the Cash column
ATM_1 <- ATM_Data %>%
  filter(ATM == 'ATM1') %>%
  summarise(ATM,Cash = sum(Cash))
```

```
ATM_1 %>%
  model(STL(Cash ~ trend() + season(window = "periodic"))) %>%
  components() %>%
    autoplot()
```



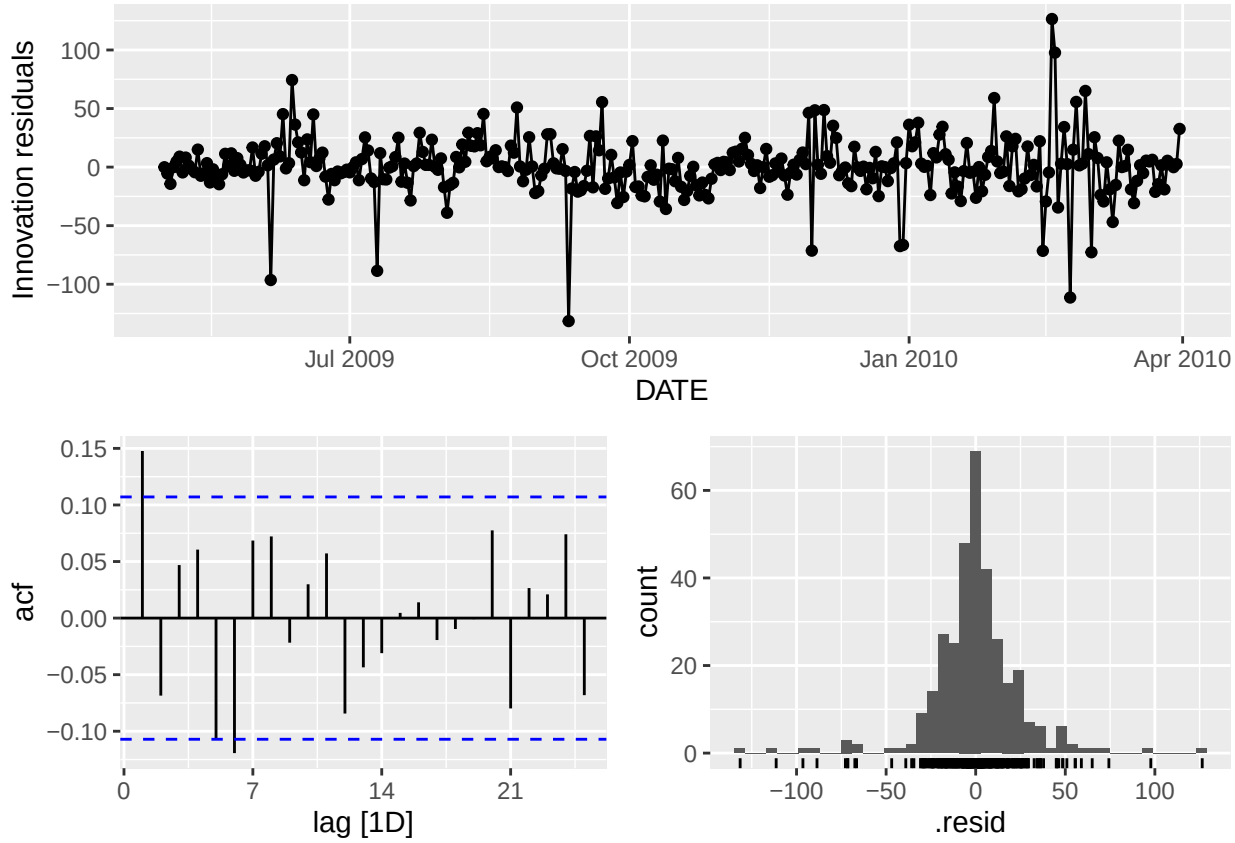

```
#splitting the data for train and test
train <- ATM_1 %>%
  filter(DATE <= as_date('2010-03-31'))

test <- ATM_1 %>%
  filter(DATE > as_date('2010-03-31'))
```

```
# fit for ETS model
ets_fit <- train %>%
  model(ETS(Cash))

# report ETS model
report(ets_fit)
```

```
## Series: Cash
## Model: ETS(A,N,A)
##   Smoothing parameters:
##     alpha = 0.0191051
##     gamma = 0.3269207
##
##   Initial states:
##      l[0]      s[0]      s[-1]    s[-2]    s[-3]    s[-4]    s[-5]    s[-6]
##  80.03069 -67.35794 -0.6711547 14.62464 10.41787 19.38412 7.530006 16.07247
##
##   sigma^2:  623.3372
```

```
##
##      AIC     AICc      BIC
## 4114.365 4115.044 4152.506
```

```
#residuals
gg_tsresiduals(ets_fit)
```



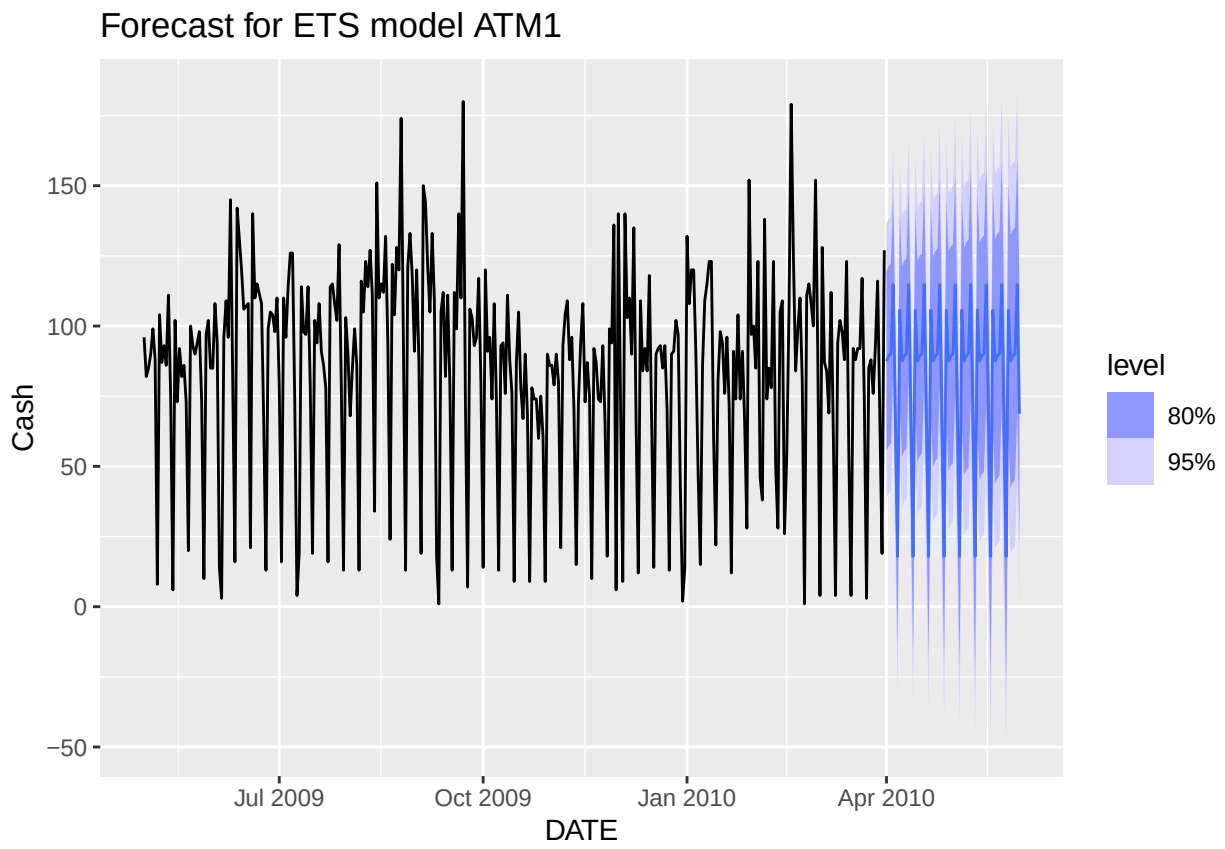

```
#Ljung box test to confirm if the data is white noise. White noise since p-value is over .05
ets_fit %>%
  augment() %>%
  features(.resid, ljung_box, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model    lb_stat lb_pvalue
##   <chr>       <dbl>     <dbl>
## 1 ETS(Cash)    35.4    0.0628
```

```
# Box Pierce test - no significant autocorrelation
ets_fit %>%
  augment() %>%
    features(.innov, box_pierce, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model    bp_stat bp_pvalue
##   <chr>       <dbl>     <dbl>
## 1 ETS(Cash)    34.3    0.0799
```

```r
# ETS forecast
ets_fc <- ets_fit %>%
  forecast(h = '2 month')

#plot ets forecast
ets_fc %>%
  autoplot(train) +
  labs(title = "Forecast for ETS model ATM1")
```



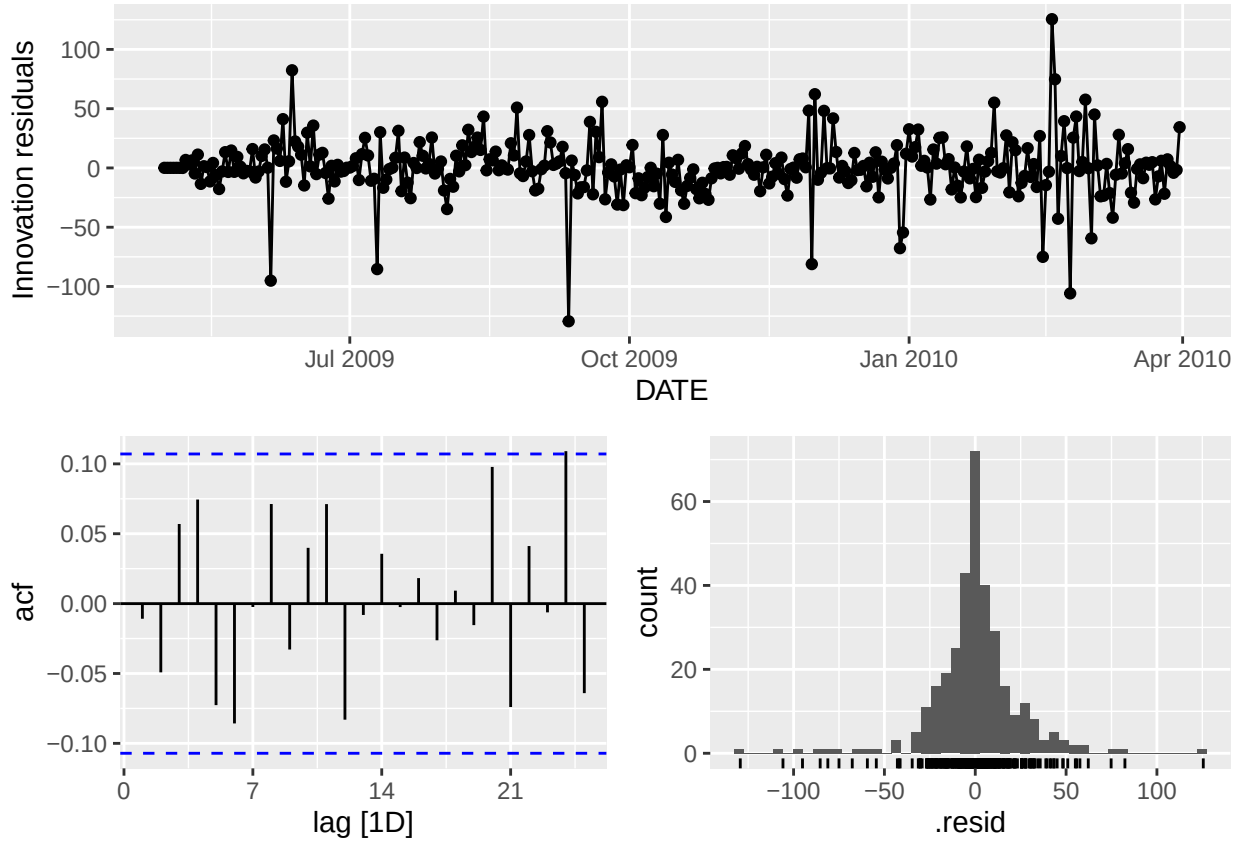

```r
# Export forecast to Excel
fc1_data <- as.data.frame(ets_fc)
write.xlsx(fc1_data, "Forecast_ATM1_FC.xlsx")
```

```r
# fit for ARIMA model
arima_fit <- train %>%
  model(ARIMA(Cash))

#report on ARIMA model
report(arima_fit)
```

```
## Series: Cash
## Model: ARIMA(0,0,1)(0,1,2)[7]
##
## Coefficients:
##          ma1     sma1     sma2
```

```
##       0.2003  -0.5834  -0.1090
## s.e.  0.0578   0.0532   0.0531
##
## sigma^2 estimated as 597.7:  log likelihood=-1514.45
## AIC=3036.9   AICc=3037.02   BIC=3052.07
```

```
# residuals
gg_tsresiduals(arima_fit)
```

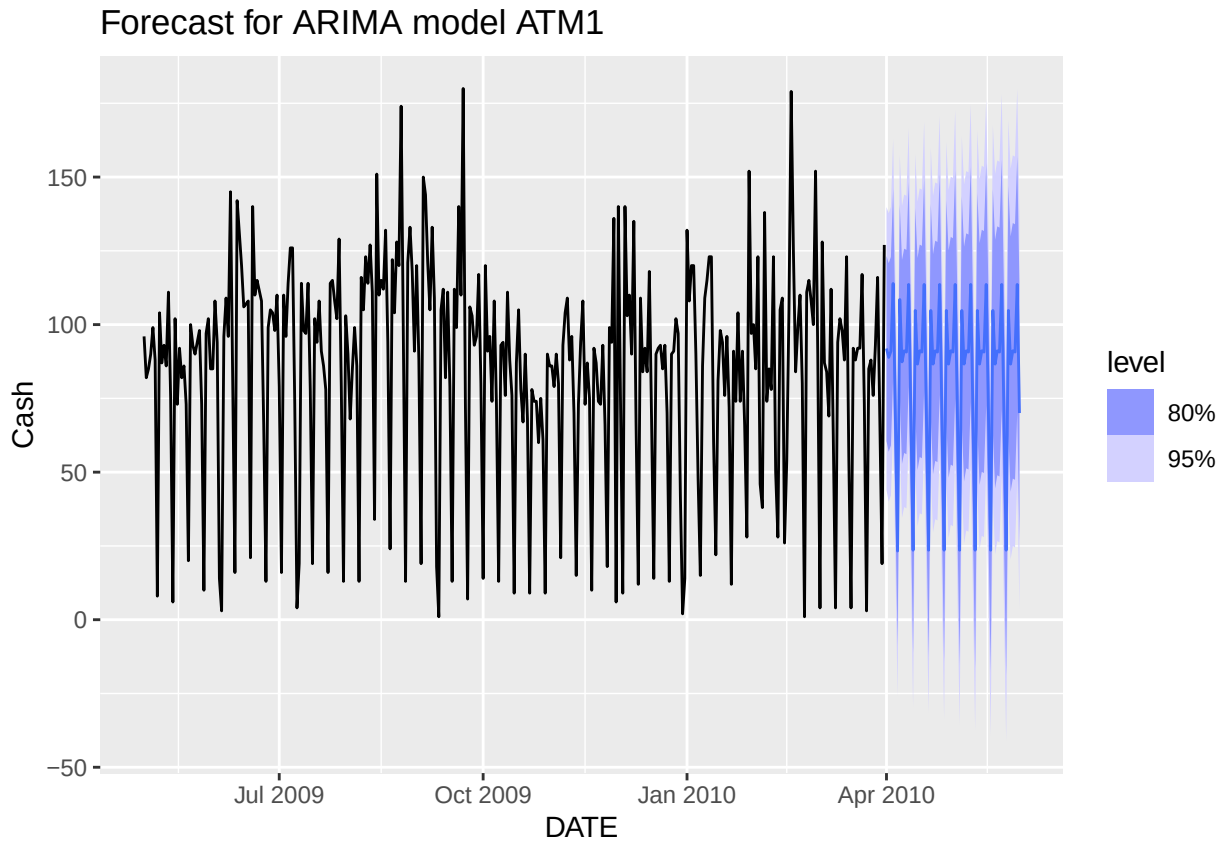



```
# Augmented Dickey-Fuller - p-value less than .05 therefore it is stationary
adf_test <- adf.test(train$Cash)
print(adf_test)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train$Cash
## Dickey-Fuller = -3.904, Lag order = 6, p-value = 0.01413
## alternative hypothesis: stationary
```

```
#Data is stationary so I can apply Arima model
# ARIMA forecast
arima_fc <- arima_fit %>%
  forecast(h ='2 month')
```

```
#plot ARIMA forecast
arima_fc %>%
  autoplot(train) +
  labs(title = "Forecast for ARIMA model ATM1")
```




```
# display Arima values
accuracy(arima_fc, test)
```

```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 31 observations are missing between 2010-05-01 and 2010-05-31
```

```
## # A tibble: 1 x 10
##   .model      .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>       <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ARIMA(Cash) Test  -6.90  12.4  9.71 -83.3  86.4   NaN   NaN -0.287
```

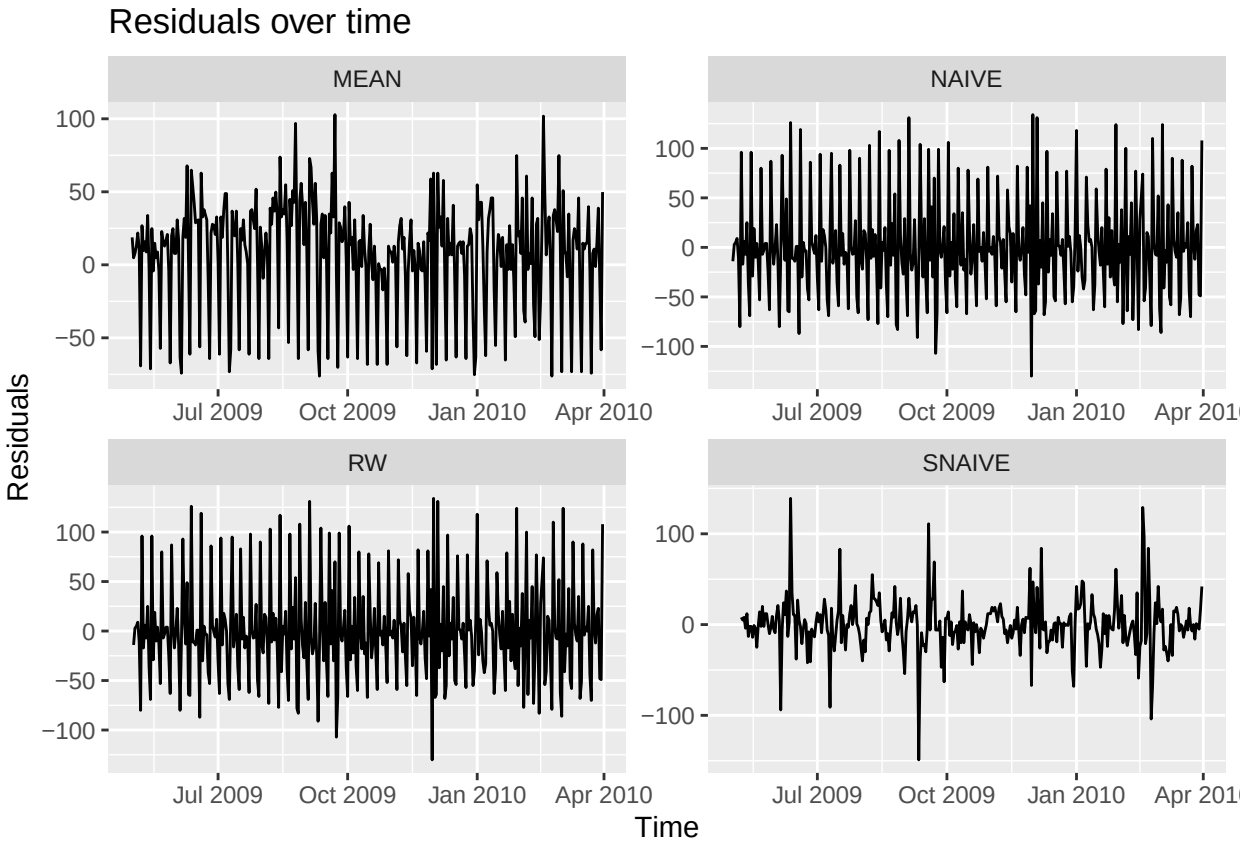
```
# display ets values
accuracy(ets_fc, test)
```

```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 31 observations are missing between 2010-05-01 and 2010-05-31
```

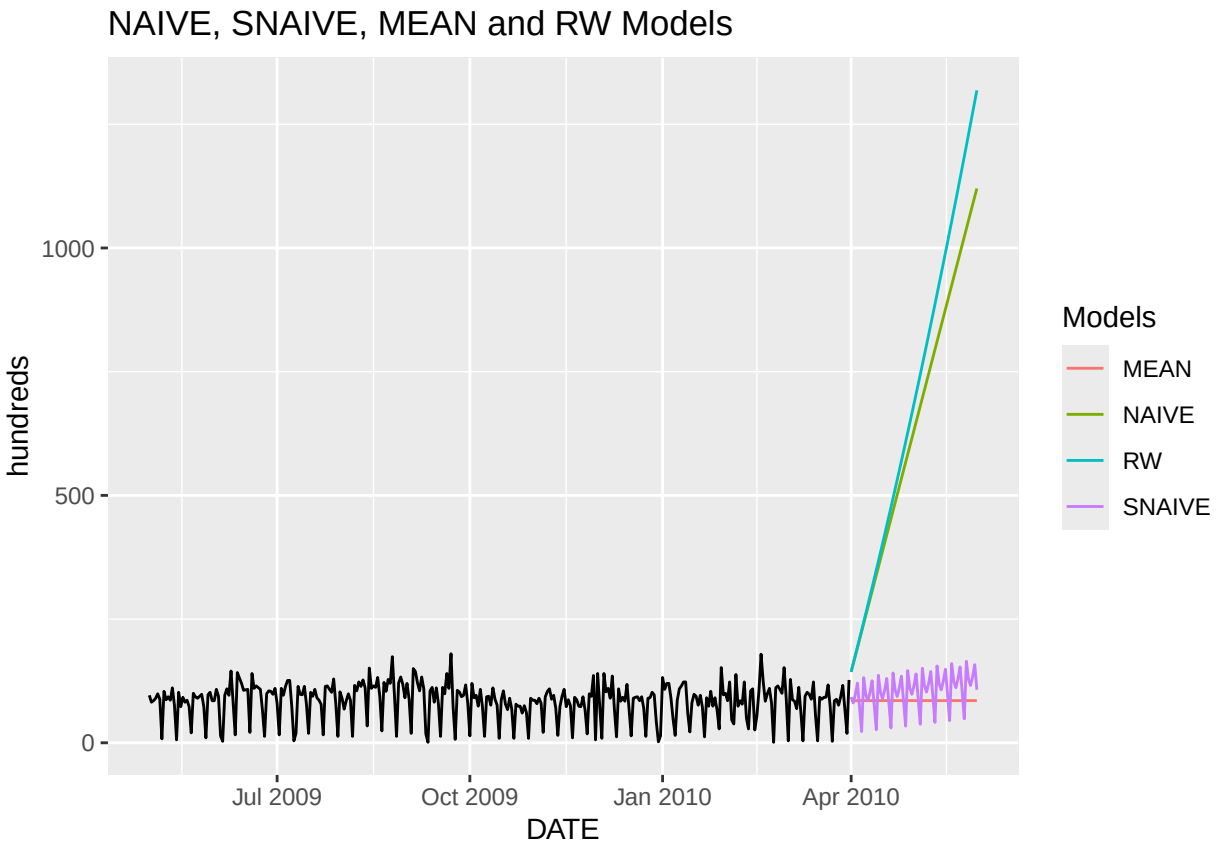
```
## # A tibble: 1 x 10
```

```
##   .model    .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>     <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ETS(Cash) Test  -5.75  11.6  9.11 -61.0  64.7   NaN   NaN -0.254
```

```r
#apply lambda
lambda <- train %>%
  features(Cash, features = guerrero) %>%
  pull(lambda_guerrero)

#box cox transformation fit
model_fit <- train %>%
  model(
    NAIVE = NAIVE(box_cox(Cash, lambda)),
    SNAIVE = SNAIVE(box_cox(Cash, lambda)),
    MEAN = MEAN(box_cox(Cash, lambda)),
    RW = RW(box_cox(Cash, lambda) ~ drift())
  )

# augment model residuals
augmented_residuals <- model_fit %>%
  augment()

# Plot residuals over time for each model
augmented_residuals %>%
  ggplot(aes(x = DATE, y = .resid)) +
  geom_line() +
  facet_wrap(~ .model, scales = "free") +
  labs(title = "Residuals over time",
       x = "Time",
       y = "Residuals")
```




```
model_fc <- model_fit %>%
  forecast(h = '2 month')

model_fc %>%
  autoplot(train, level = NULL) +
  labs(title = "NAIVE, SNAIVE, MEAN and RW Models",  y = 'hundreds') +
  guides(colour = guide_legend(title = "Models"))
```



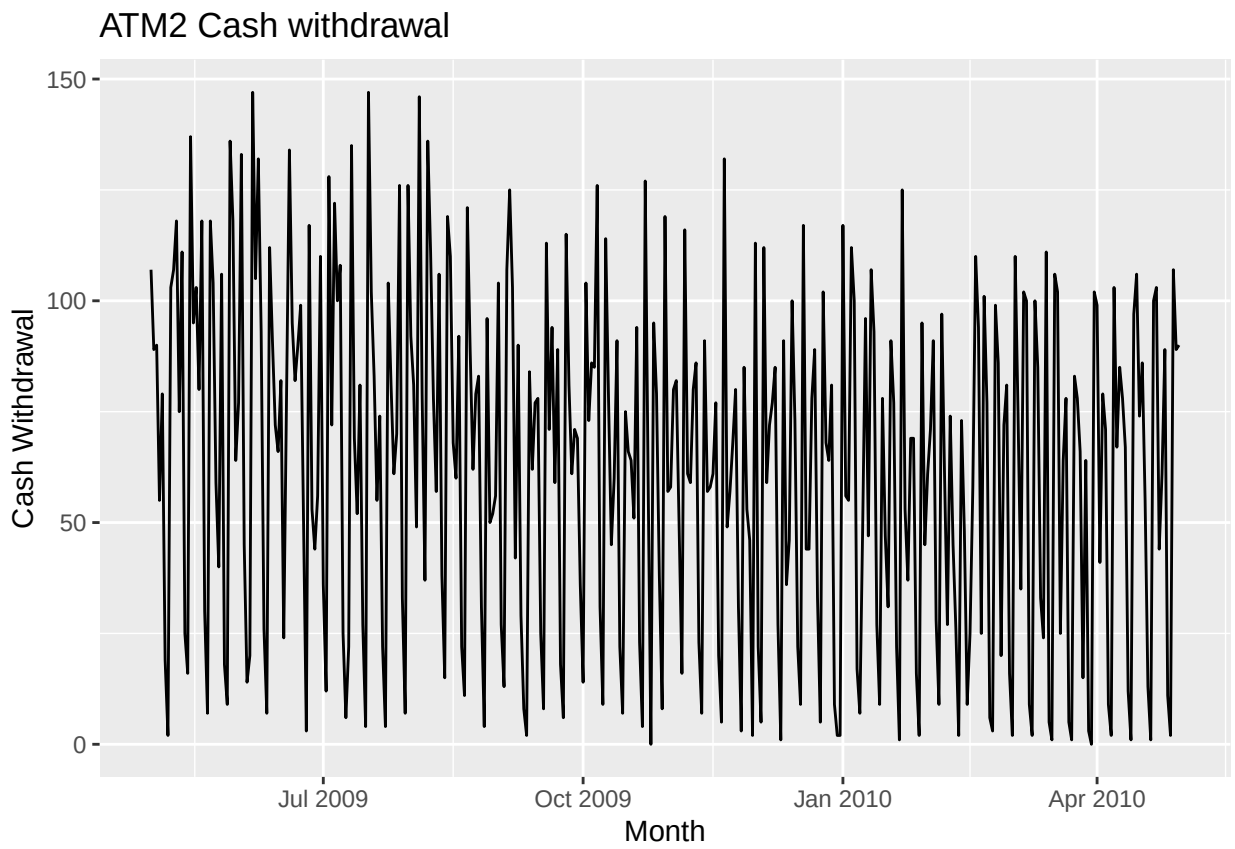

```
#accuracy on model - lowest MAE, RMSE or MAPE
accuracy(model_fc, test)
```

```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 31 observations are missing between 2010-05-01 and 2010-05-31
```

```
## # A tibble: 4 x 10
##   .model .type      ME  RMSE   MAE    MPE  MAPE  MASE RMSSE   ACF1
##   <chr>  <chr>   <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 MEAN   Test    -8.57  32.0  19.4  -317.  328.   NaN   NaN -0.147
## 2 NAIVE  Test  -303.   337.  303.  -1821. 1821.   NaN   NaN  0.855
## 3 RW     Test  -321.   360.  321.  -1913. 1913.   NaN   NaN  0.863
## 4 SNAIVE Test   -18.8   24.7  21.1  -114.  117.   NaN   NaN  0.228
```

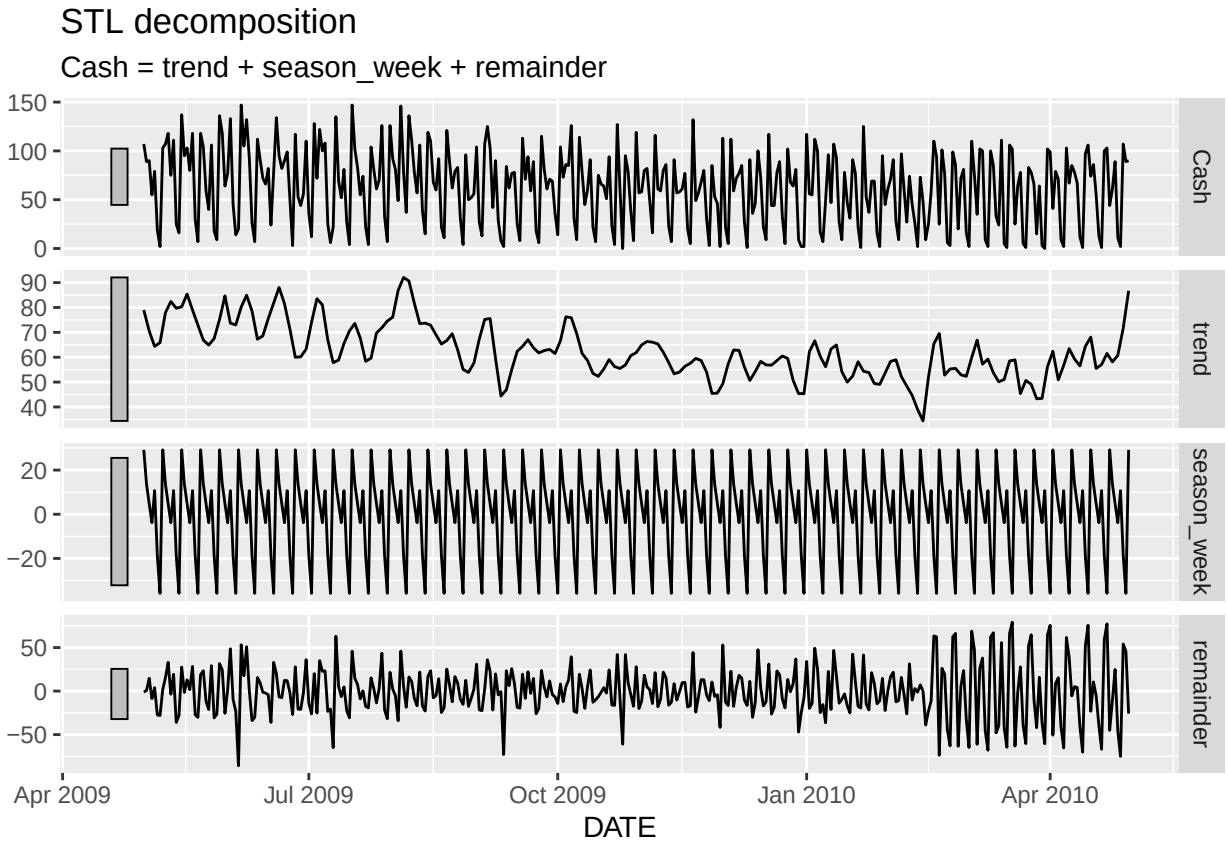
ATM2 ATM2 Data Idecided to check if there is white noise on it to run an ETS model. After reviewing the dataset, I did not find white noise, even after conducting the box-cox transformation the p-value is less than .05. The use of the ETS model is not recommend ti for this data. I ran the ARIMA model and with the ADF test with less than .05, I can tell that the data is stationary to run the forecast. Overall the best model to use for this data is using the ARIMA model.

```
ATM_Data %>%
  filter(ATM == 'ATM2') %>%
  autoplot(Cash) +
  labs(title = "ATM2 Cash withdrawal",x = "Month", y = "Cash Withdrawal")
```




```
ATM_2 <- ATM_Data %>%
  filter(ATM == 'ATM2') %>%
  summarise(ATM,Cash = sum(Cash))
```

```
ATM_2 %>%
  model(STL(Cash ~ trend() + season(window = "periodic"))) %>%
  components() %>%
    autoplot()
```



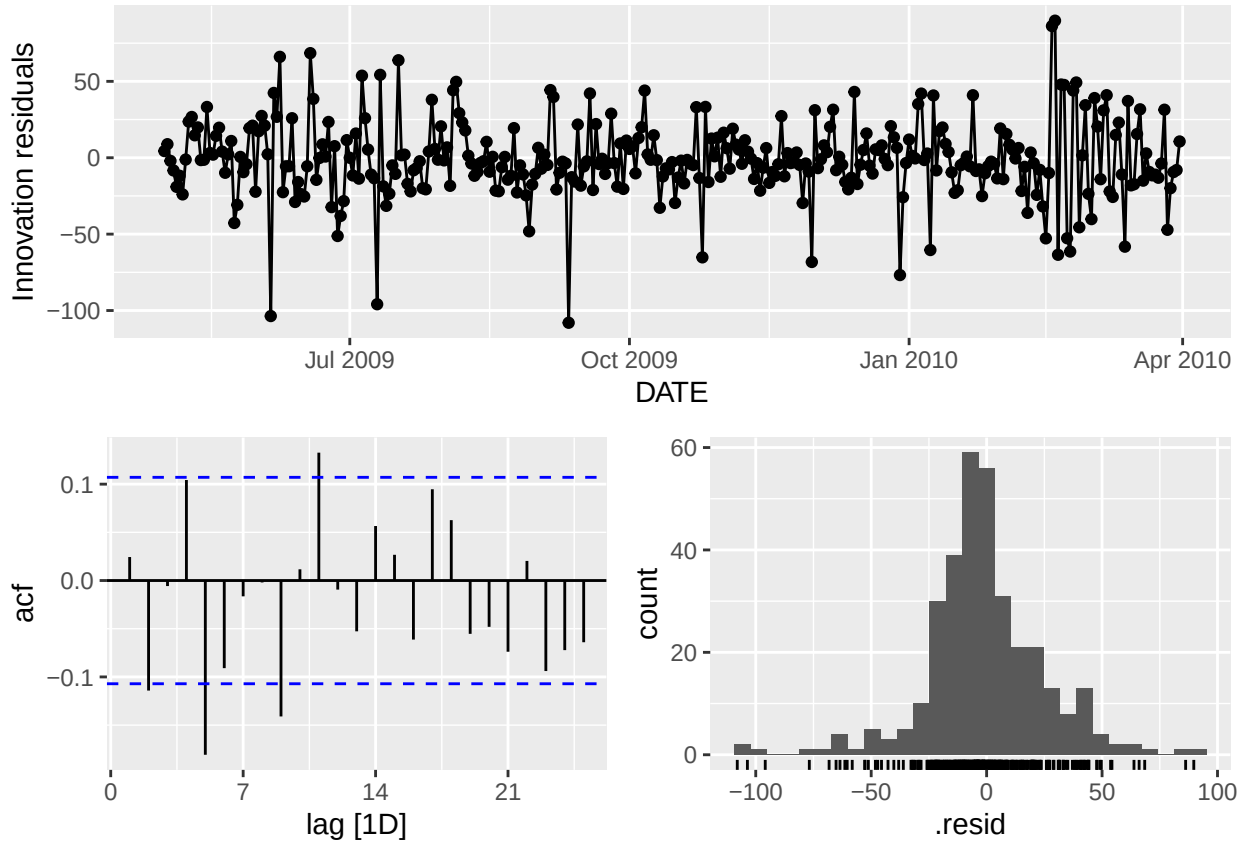

```
#splitting the data
train2 <- ATM_2 %>%
  filter(DATE <= as_date('2010-03-31'))

test2 <- ATM_2 %>%
  filter(DATE > as_date('2010-03-31'))
```

```
# fit for ETS model
ets_fit2 <- train2 %>%
  model(ETS(Cash))

# report ETS model
report(ets_fit2)
```

```
## Series: Cash
## Model: ETS(A,N,A)
##   Smoothing parameters:
##     alpha = 0.0001000115
##     gamma = 0.3618479
##
##   Initial states:
##      l[0]      s[0]     s[-1]     s[-2]     s[-3]     s[-4]     s[-5]     s[-6]
##  70.42189 -44.42227 -39.62642 27.61801 -7.098018 21.63878 9.701075 32.18883
##
##   sigma^2:  675.5735
```

```
##
##      AIC     AICc      BIC
## 4141.324 4142.003 4179.465
```

```
#residuals
gg_tsresiduals(ets_fit2)
```

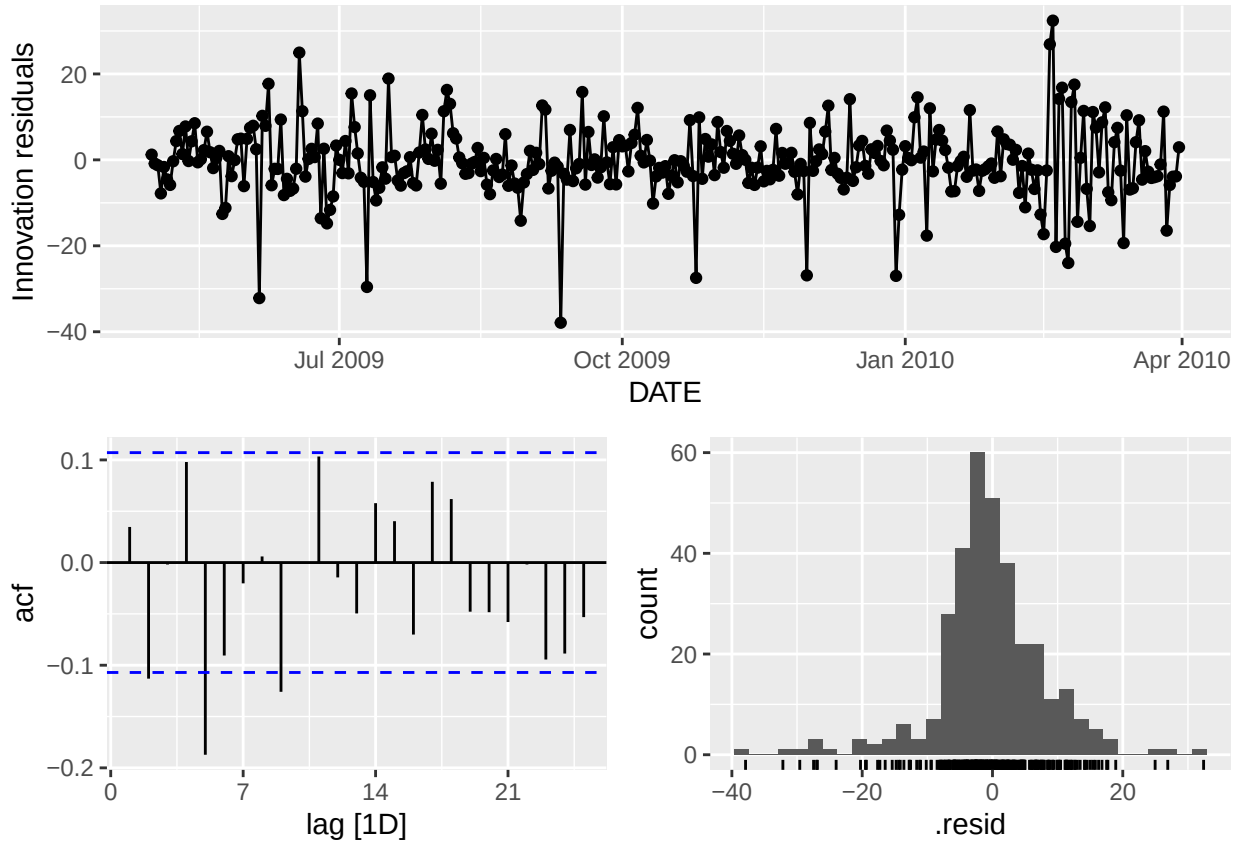



```
#Ljung box test- p-value is less than .05
ets_fit2 %>%
  augment() %>%
  features(.resid, ljung_box, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model    lb_stat lb_pvalue
##   <chr>       <dbl>     <dbl>
## 1 ETS(Cash)    52.9  0.000593
```

```
# Box-Cox Transformation
lambda2 <- train2 %>%
  features(Cash, features = guerrero) %>%
  pull(lambda_guerrero)

# fit for ETS model with box_cox transformation
ets_fit2 <- train2 %>%
  model(ETS(box_cox(Cash, lambda2)))
```

```
#residuals
gg_tsresiduals(ets_fit2)
```




```
#Ljung box test- p-value below .05 but better than previously
ets_fit2 %>%
  augment() %>%
  features(.resid, ljung_box, lag = 24)
```

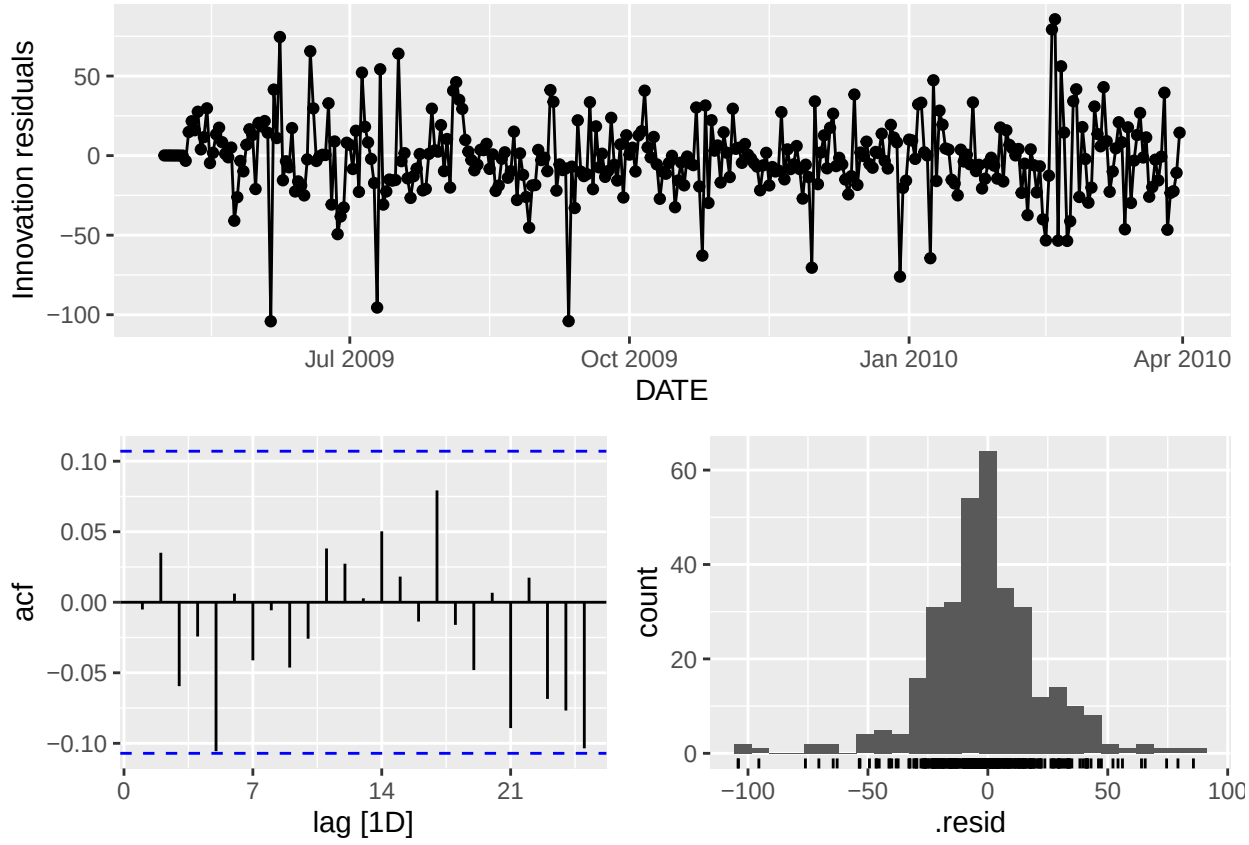
```
## # A tibble: 1 x 3
##   .model                       lb_stat lb_pvalue
##   <chr>                          <dbl>     <dbl>
## 1 ETS(box_cox(Cash, lambda2))    47.1   0.00327
```

```
# fit for ARIMA model
arima_fit2 <- train2 %>%
  model(ARIMA(Cash))

#report on ARIMA model
report(arima_fit2)
```

```
## Series: Cash
## Model: ARIMA(2,0,2)(0,1,1)[7]
##
```

```
## Coefficients:
##           ar1      ar2     ma1     ma2    sma1
##       -0.4322  -0.9259  0.4799  0.8085  -0.779
## s.e.   0.0484   0.0399  0.0805  0.0537   0.042
##
## sigma^2 estimated as 626.2:  log likelihood=-1521.62
## AIC=3055.25   AICc=3055.51   BIC=3078.01
```

```
# residuals
gg_tsresiduals(arima_fit2)
```



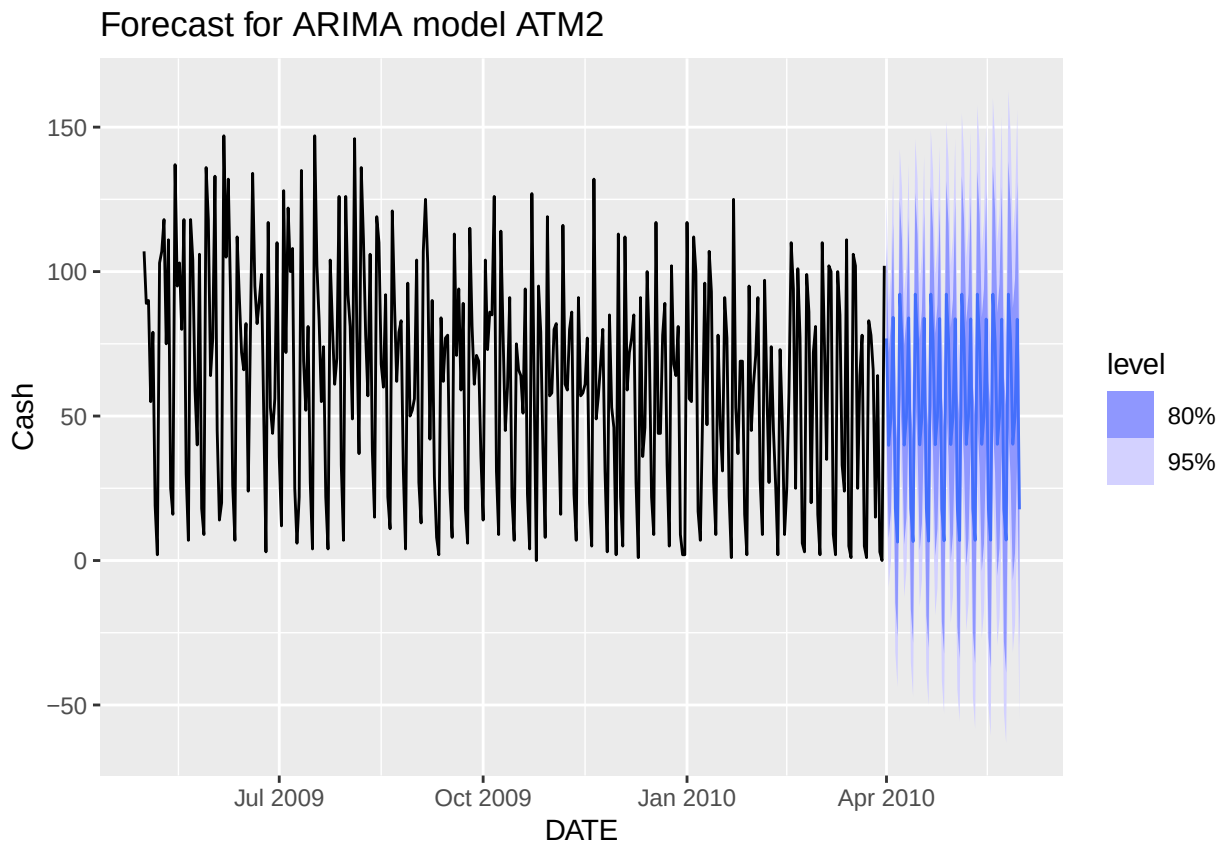

```
adf_test2 <- adf.test(train2$Cash)
```

```
## Warning in adf.test(train2$Cash): p-value smaller than printed p-value
```

```
print(adf_test2)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train2$Cash
## Dickey-Fuller = -6.2416, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
# ARIMA forecast
arima_fc2 <- arima_fit2 %>%
  forecast(h ='2 month')

#plot ARIMA forecast
arima_fc2 %>%
  autoplot(train2) +
  labs(title = "Forecast for ARIMA model ATM2")
```



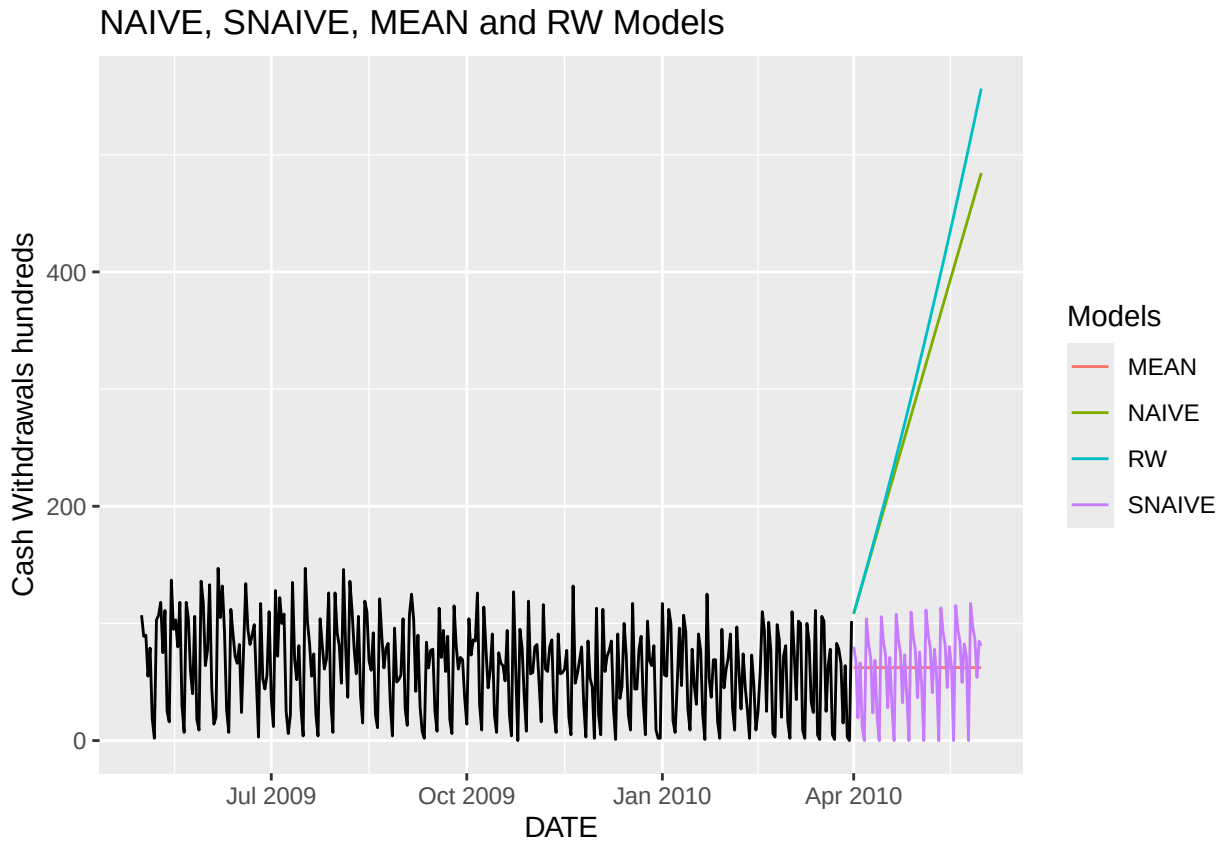

```
# Export forecast to Excel
fc2_data <- as.data.frame(arima_fc2)
write.xlsx(fc2_data, "Forecast_ATM2_FC.xlsx")
```

```
# display Arima Values
accuracy(arima_fc2, test2)
```

```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 31 observations are missing between 2010-05-01 and 2010-05-31
```

```
## # A tibble: 1 x 10
##   .model      .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>       <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ARIMA(Cash) Test   8.62  21.0  16.3 -50.5  79.5   NaN   NaN 0.0999
```

```r
#plot to compare between models
model_fit2 <- train2 %>%
  model(
    NAIVE = NAIVE(box_cox(Cash, lambda2)),
    SNAIVE = SNAIVE(box_cox(Cash, lambda2)),
    MEAN = MEAN(box_cox(Cash, lambda2)),
    RW = RW(box_cox(Cash, lambda2) ~ drift())
  )

model_fc2 <- model_fit2 %>%
  forecast(h = '2 month')

model_fc2 %>%
  autoplot(train2, level = NULL) +
  labs(title = "NAIVE, SNAIVE, MEAN and RW Models",  y = 'Cash Withdrawals hundreds') +
  guides(colour = guide_legend(title = "Models"))
```




```r
#accuracy on model - lowest MAE, RMSE or MAPE
accuracy(model_fc2, test2)
```

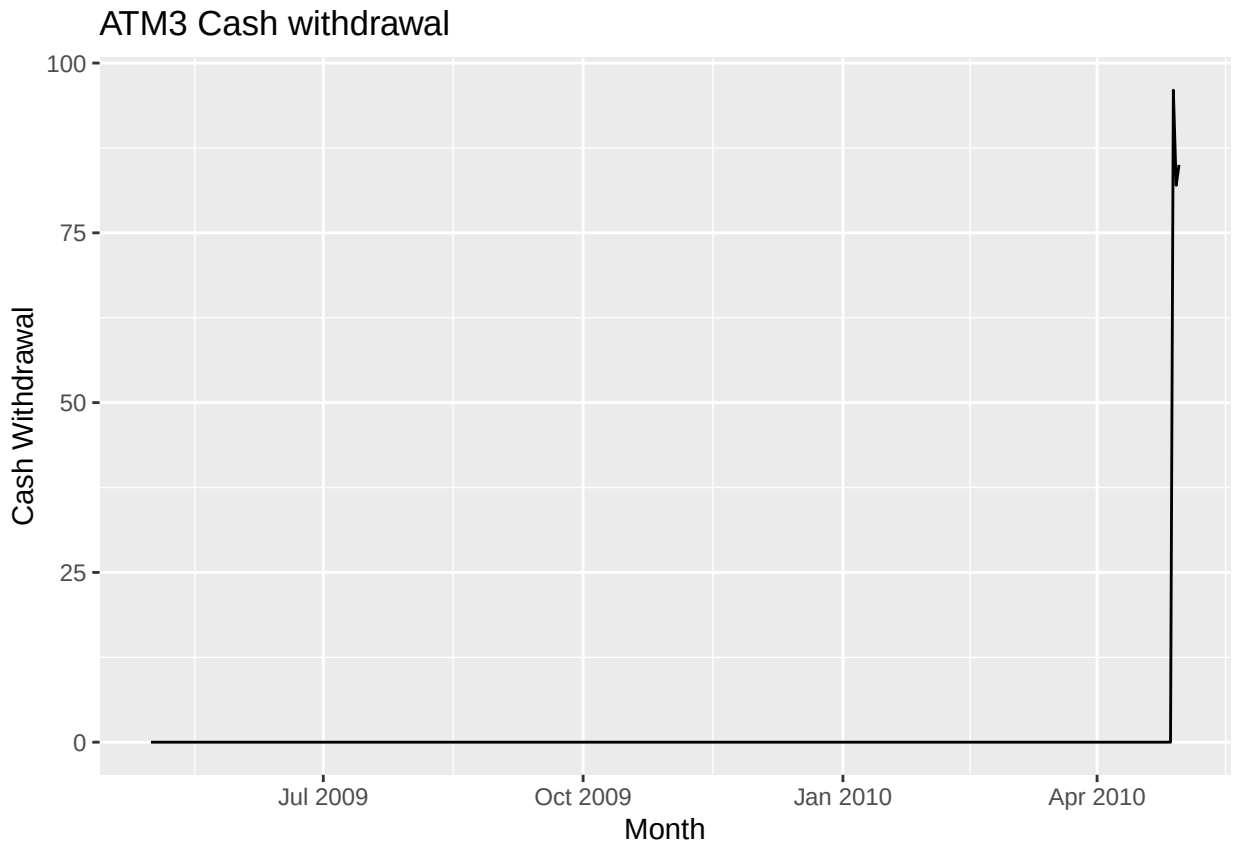
```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 31 observations are missing between 2010-05-01 and 2010-05-31
```

```
## # A tibble: 4 x 10
```

```
##   .model .type       ME  RMSE   MAE      MPE   MAPE  MASE RMSSE   ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl>    <dbl>  <dbl> <dbl> <dbl>  <dbl>
## 1 MEAN   Test    -0.918  37.1  32.1  -659.    692.    NaN   NaN  0.136
## 2 NAIVE  Test  -138.    152.  138.  -2379.   2379.    NaN   NaN  0.634
## 3 RW     Test  -144.    159.  144.  -2453.   2453.    NaN   NaN  0.667
## 4 SNAIVE Test     4.93   23.0  15.9     3.22   48.3   NaN   NaN -0.288
```

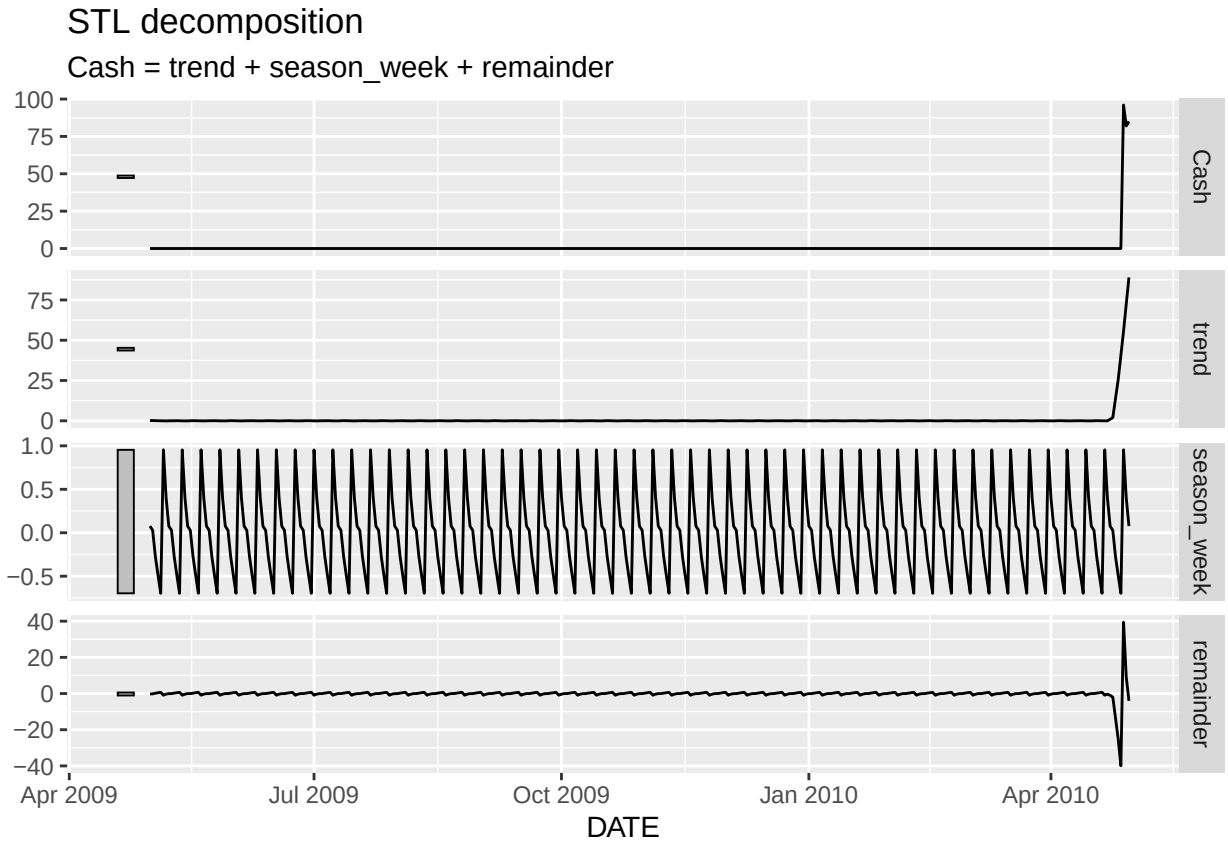
ATM3 The data for ATM3 is mostly with zero values, only a few data in April was available to work with. I decided that the best forecast is the seasonal naive based on the charts without splitting up the data for test and train .Splitting up the data for train data prior to April would not provide much insight as the forecast would just be zero and can not predict that the number would increase after March.

```
ATM_Data %>%
  filter(ATM == 'ATM3') %>%
  autoplot(Cash) +
  labs(title = "ATM3 Cash withdrawal",x = "Month", y = "Cash Withdrawal")
```




```
ATM_3 <- ATM_Data %>%
  filter(ATM == 'ATM3') %>%
  summarise(ATM,Cash = sum(Cash))
```

```
ATM_3 %>%
  model(STL(Cash ~ trend() + season(window = "periodic"))) %>%
  components() %>%
    autoplot()
```




```
#splitting the data
train3 <- ATM_3 %>%
  filter(DATE <= as_date('2010-03-31'))

test3 <- ATM_3 %>%
  filter(DATE > as_date('2010-03-31'))
```

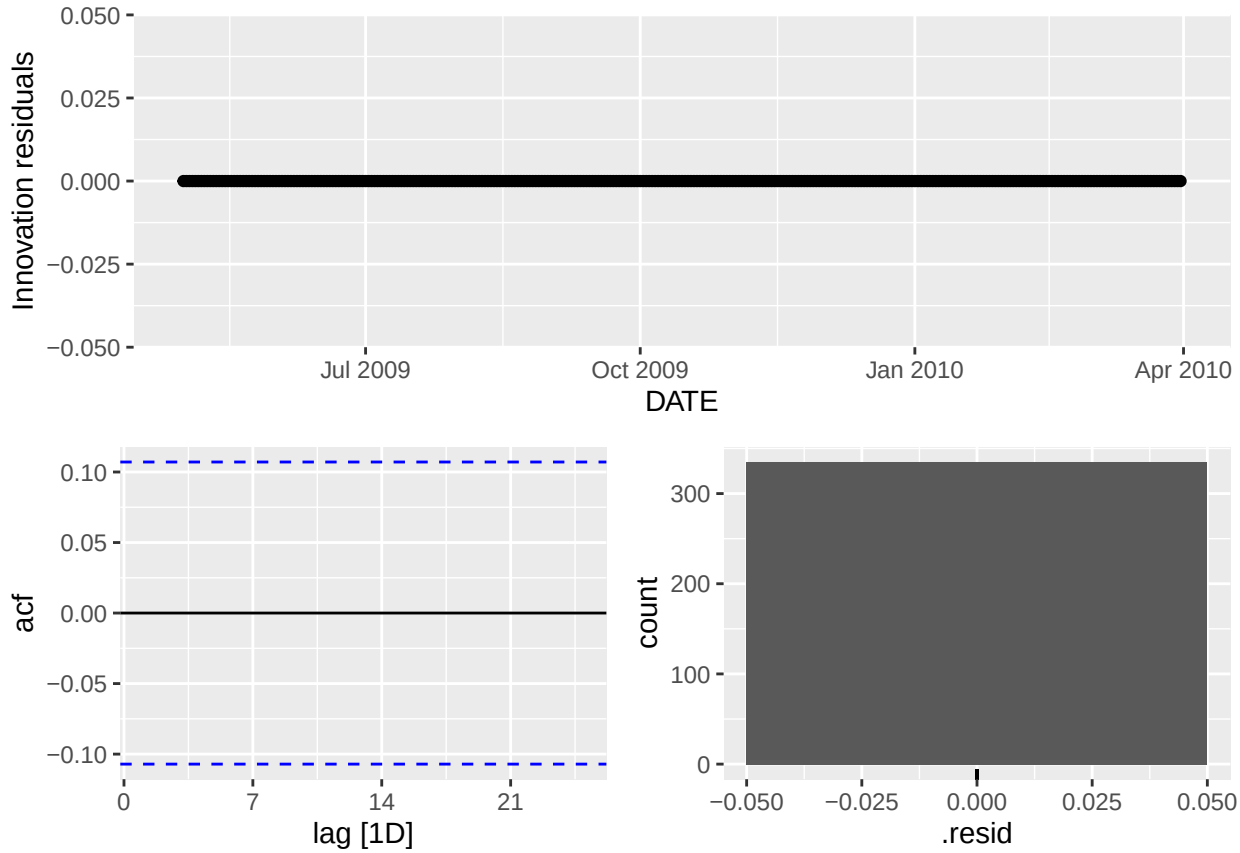
```
# fit for ETS model
ets_fit3 <- train3 %>%
  model(ETS(Cash))

# report ETS model
report(ets_fit3)
```

```
## Series: Cash
## Model: ETS(A,N,N)
##   Smoothing parameters:
##     alpha = 0.20006
##
##   Initial states:
##  l[0]
##     0
##
##   sigma^2:  0
##
```

```
##  AIC AICc  BIC
## -Inf -Inf -Inf
```

```
#residuals
gg_tsresiduals(ets_fit3)
```

```
## Warning: Removed 25 rows containing missing values or values outside the scale range
## (`geom_segment()`).
```

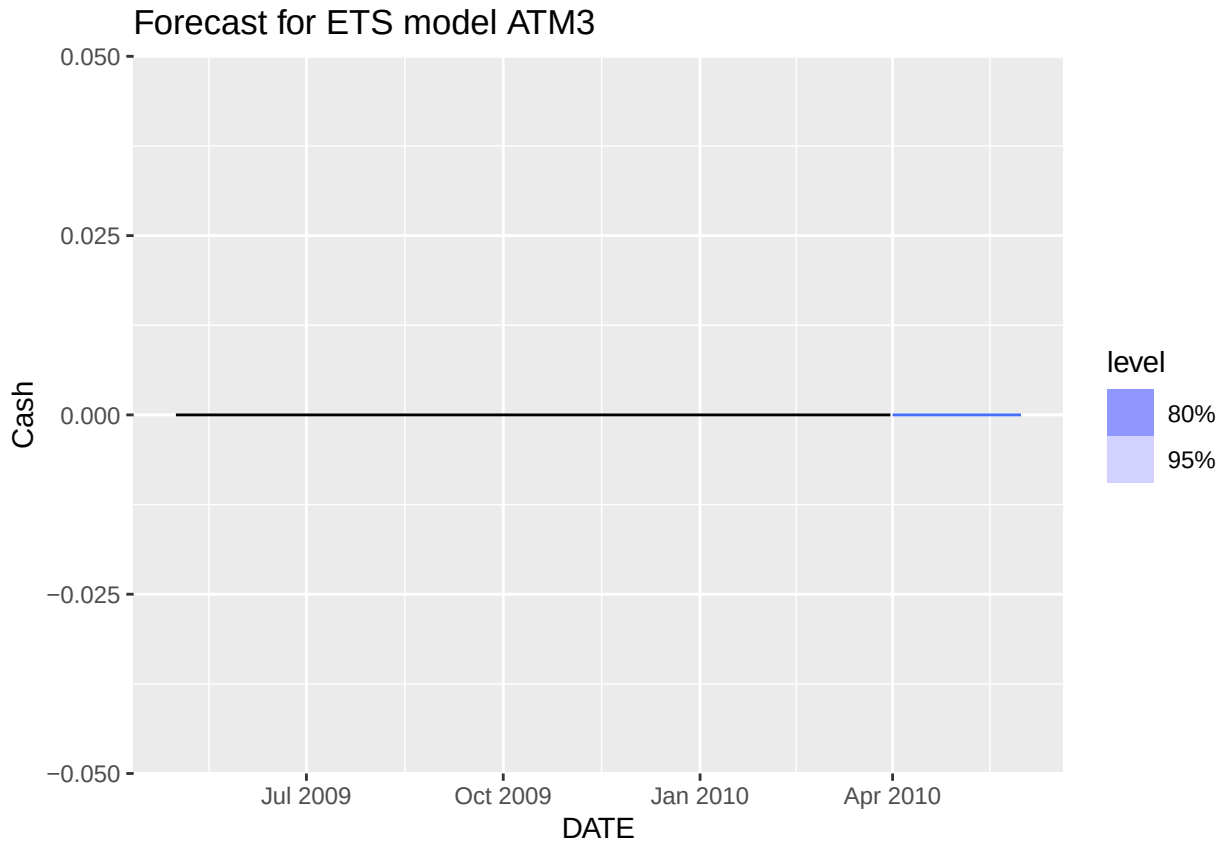



```
#Ljung box test- White noise or not
ets_fit3 %>%
  augment() %>%
  features(.resid, ljung_box, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model    lb_stat lb_pvalue
##   <chr>       <dbl>     <dbl>
## 1 ETS(Cash)     NaN       NaN
```

```
# ETS forecast
ets_fc3 <- ets_fit3 %>%
  forecast(h = '2 month')

#plot ets forecast
```

```
ets_fc3 %>%
  autoplot(train3) +
  labs(title = "Forecast for ETS model ATM3")
```

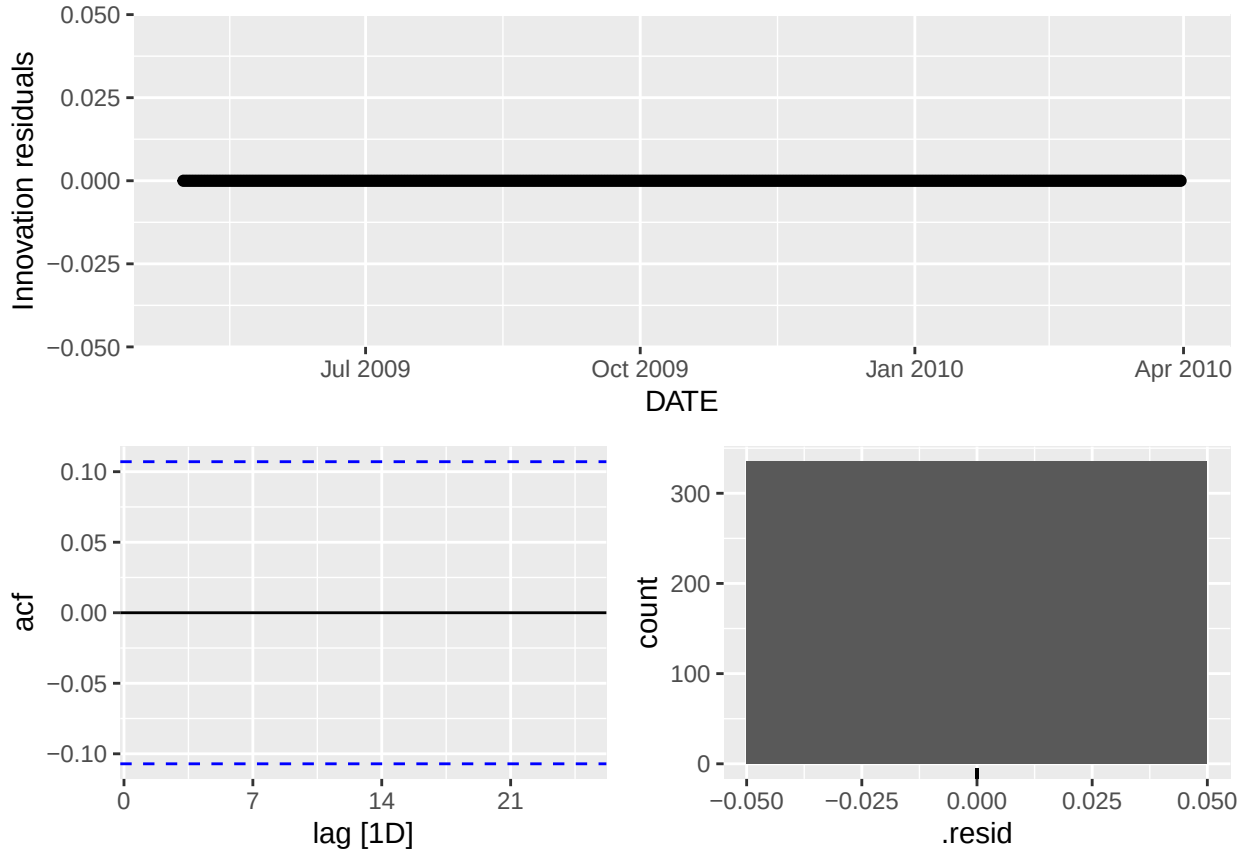



```
# fit for ARIMA model
arima_fit3 <- train3 %>%
  model(ARIMA(Cash))

#report on ARIMA model
report(arima_fit3)
```

```
## Series: Cash
## Model: ARIMA(0,0,0)(0,0,1)[7]
##
## sigma^2 estimated as 0:  log likelihood=Inf
## AIC=-Inf   AICc=-Inf   BIC=-Inf
```

```
# residuals
gg_tsresiduals(arima_fit3)
```

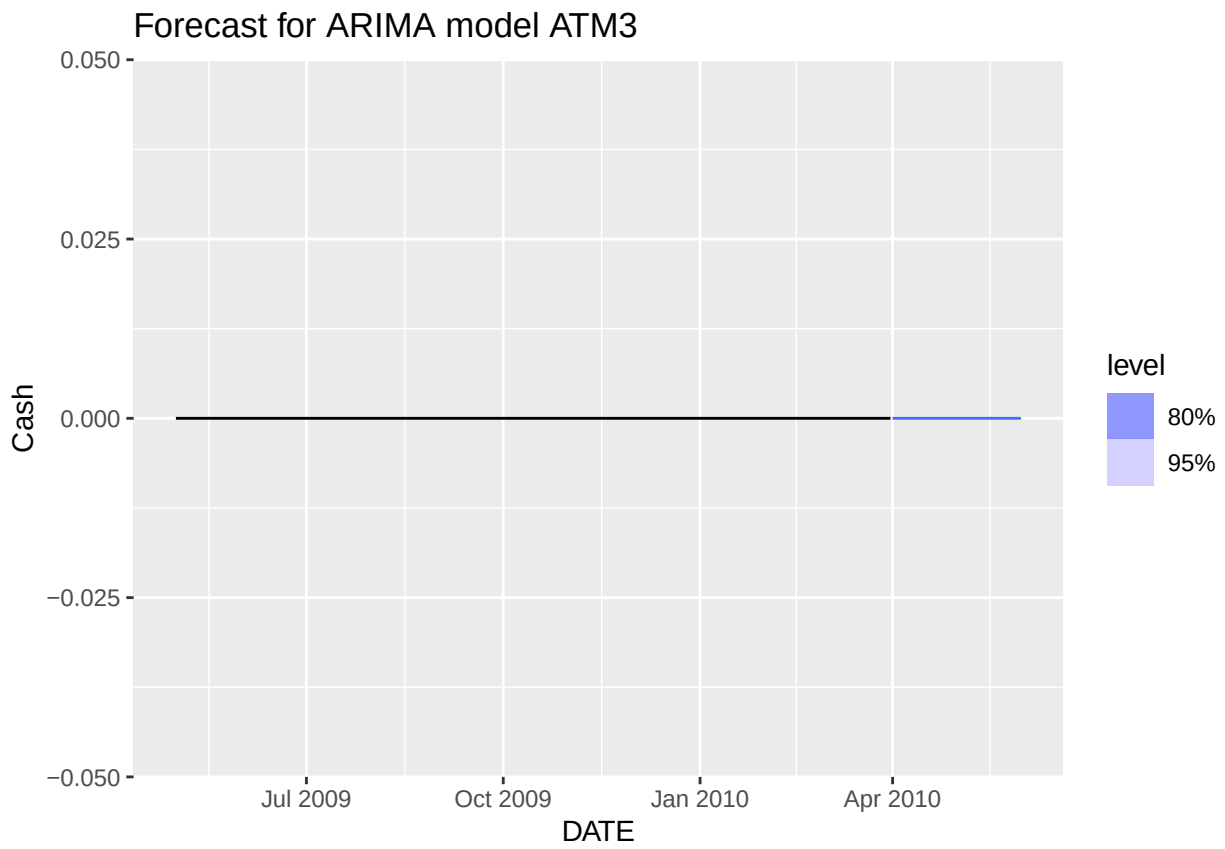
```
## Warning: Removed 25 rows containing missing values or values outside the scale range
## (`geom_segment()`).
```




```
# Augmented Dickey-Fuller
adf_test3 <- adf.test(train3$Cash)
print(adf_test3)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train3$Cash
## Dickey-Fuller = NaN, Lag order = 6, p-value = NA
## alternative hypothesis: stationary
```

```
# ARIMA forecast
arima_fc3 <- arima_fit3 %>%
  forecast(h ='2 month')

#plot ARIMA forecast
arima_fc3 %>%
  autoplot(train3) +
  labs(title = "Forecast for ARIMA model ATM3")
```




```
# Export forecast to Excel
fc3_data <- as.data.frame(arima_fc3)
write.xlsx(fc3_data, "Forecast_ATM3_FC.xlsx")
```

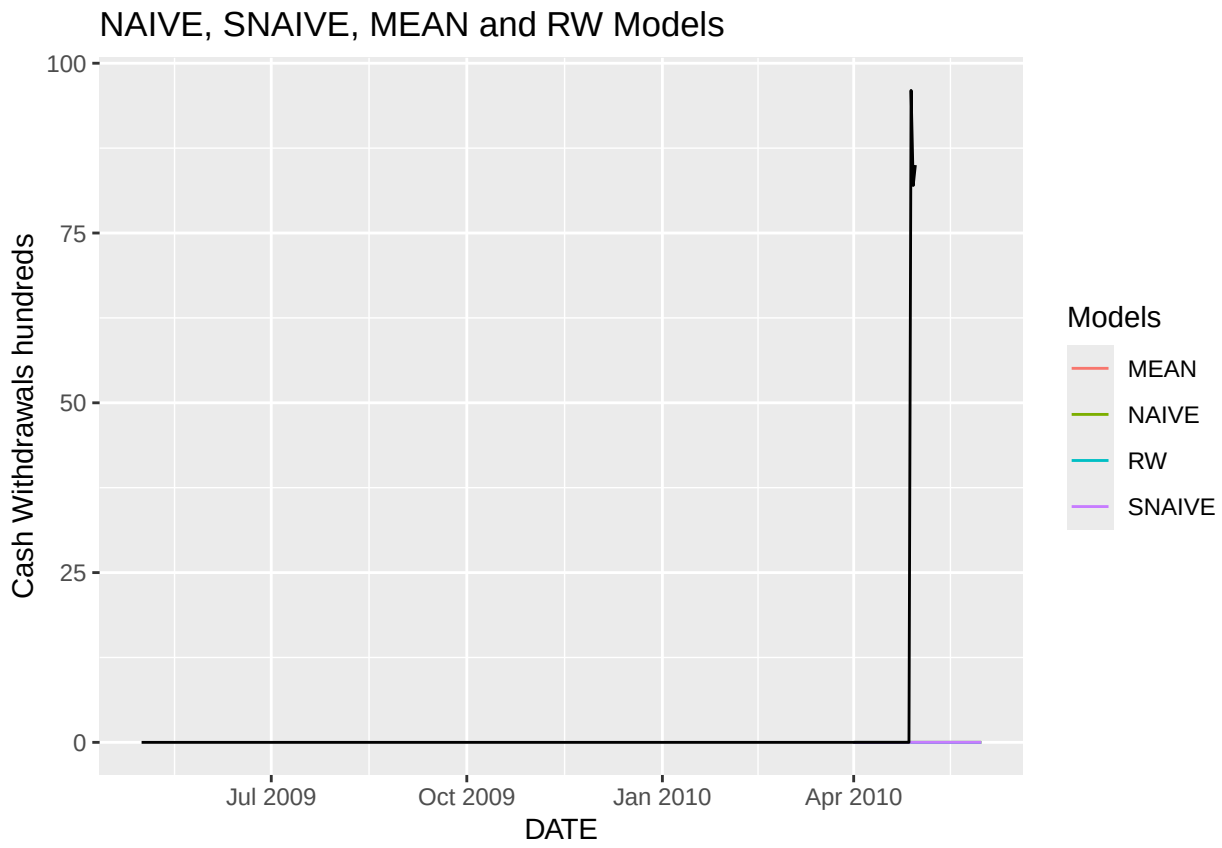
```
# Accuracy of ARIMA forecast - MAE and RMSE
accuracy(arima_fc3, test3)
```

```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 31 observations are missing between 2010-05-01 and 2010-05-31
```

```
## # A tibble: 1 x 10
##   .model      .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>       <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ARIMA(Cash) Test   8.77  27.8  8.77   100   100   NaN   NaN 0.633
```

```
# plot to compare between models
model_fit3 <- train3 %>%
  model(
    NAIVE = NAIVE(Cash),
    SNAIVE = SNAIVE(Cash),
    MEAN = MEAN(Cash),
    RW = RW(Cash ~ drift())
  )

model_fc3 <- model_fit3 %>%
```

```
  forecast(h = '2 month')

model_fc3 %>%
  autoplot(ATM_3, level = NULL) +
  labs(title = "NAIVE, SNAIVE, MEAN and RW Models",  y = 'Cash Withdrawals hundreds') +
  guides(colour = guide_legend(title = "Models"))
```



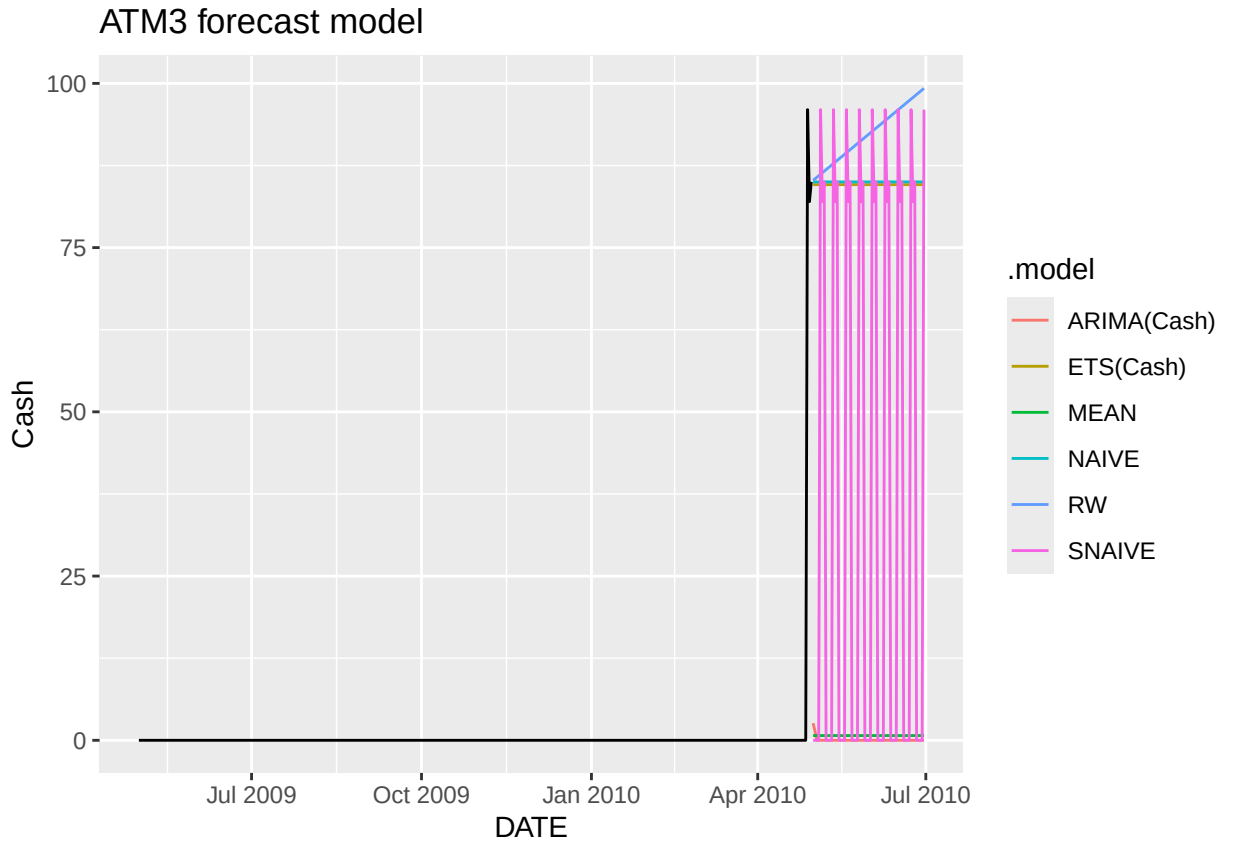

```
#accuracy on model - lowest MAE, RMSE or MAPE
accuracy(model_fc3, test3)
```

```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 31 observations are missing between 2010-05-01 and 2010-05-31
```

```
## # A tibble: 4 x 10
##   .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 MEAN   Test   8.77  27.8  8.77   100   100   NaN   NaN 0.633
## 2 NAIVE  Test   8.77  27.8  8.77   100   100   NaN   NaN 0.633
## 3 RW     Test   8.77  27.8  8.77   100   100   NaN   NaN 0.633
## 4 SNAIVE Test   8.77  27.8  8.77   100   100   NaN   NaN 0.633
```

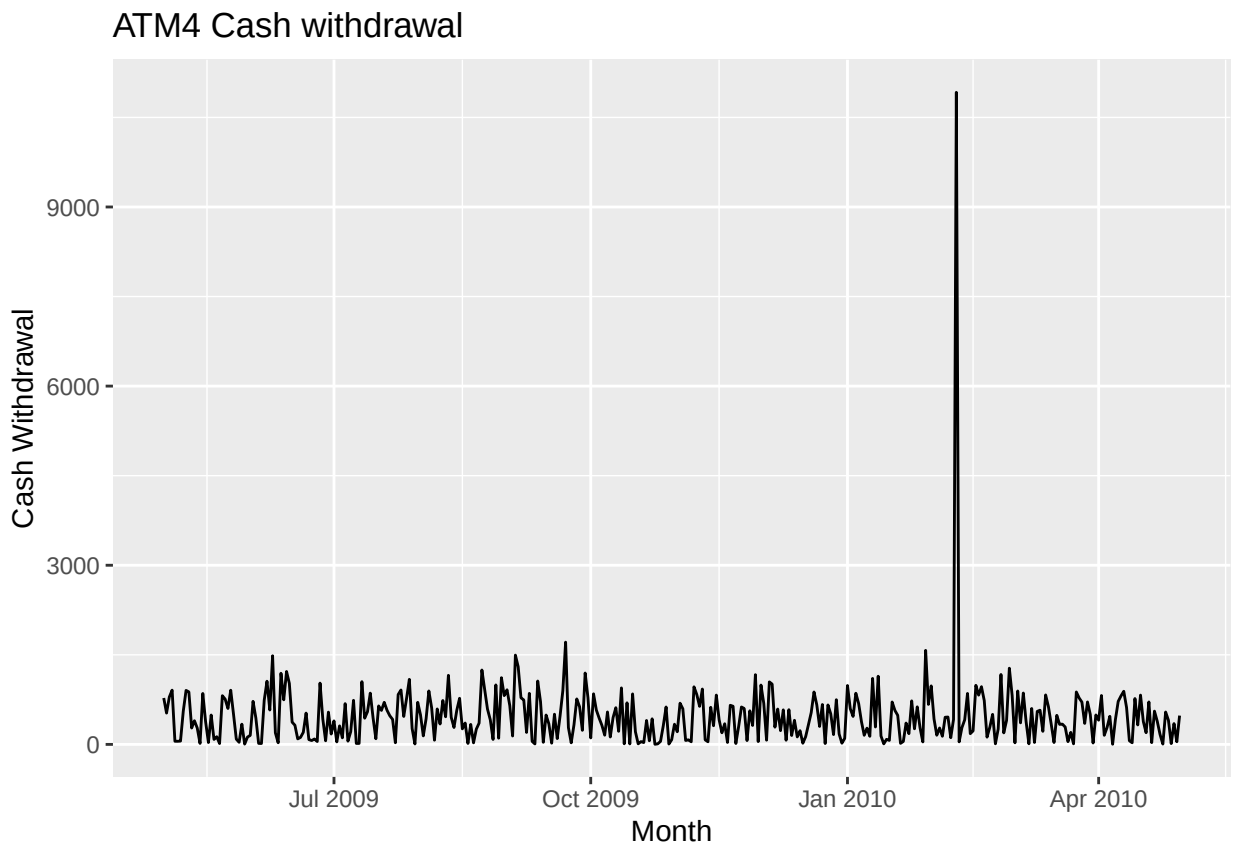
```
# run forecast model without splitting data
ATM_3 %>%
  model(
```

```
    ETS(Cash),
    ARIMA(Cash),
    NAIVE = NAIVE(Cash),
    SNAIVE = SNAIVE(Cash),
    MEAN = MEAN(Cash),
    RW = RW(Cash ~ drift())
        )  %>%
  forecast(h = '2 month') %>%
  autoplot(ATM_3, level = NULL) +
labs(title = "ATM3 forecast model")
```



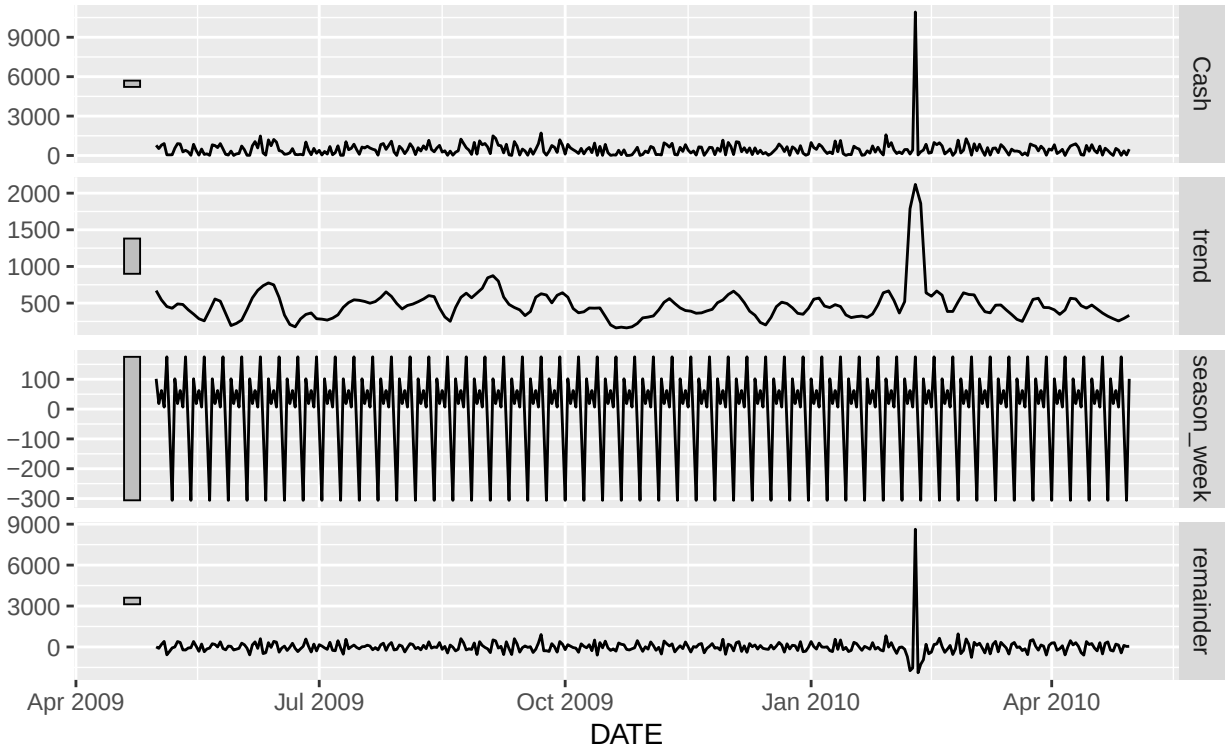
ATM4 For the ATM4 data, I decided to use a ETS model to see if the data has white noise on it, before running the forecast. The ARIMA model was applied to find out if the data is stationary or not. After comparing between models, I decided to use the Random Walk model since it has the best RMSE compared to all the different models.

```
ATM_Data %>%
  filter(ATM == 'ATM4') %>%
  autoplot(Cash) +
  labs(title = "ATM4 Cash withdrawal",x = "Month", y = "Cash Withdrawal")
```




```
ATM_4 <- ATM_Data %>%
  filter(ATM == 'ATM4') %>%
  summarise(ATM,Cash = sum(Cash))
```

```
ATM_4 %>%
  model(STL(Cash ~ trend() + season(window = "periodic"))) %>%
  components() %>%
    autoplot()
```



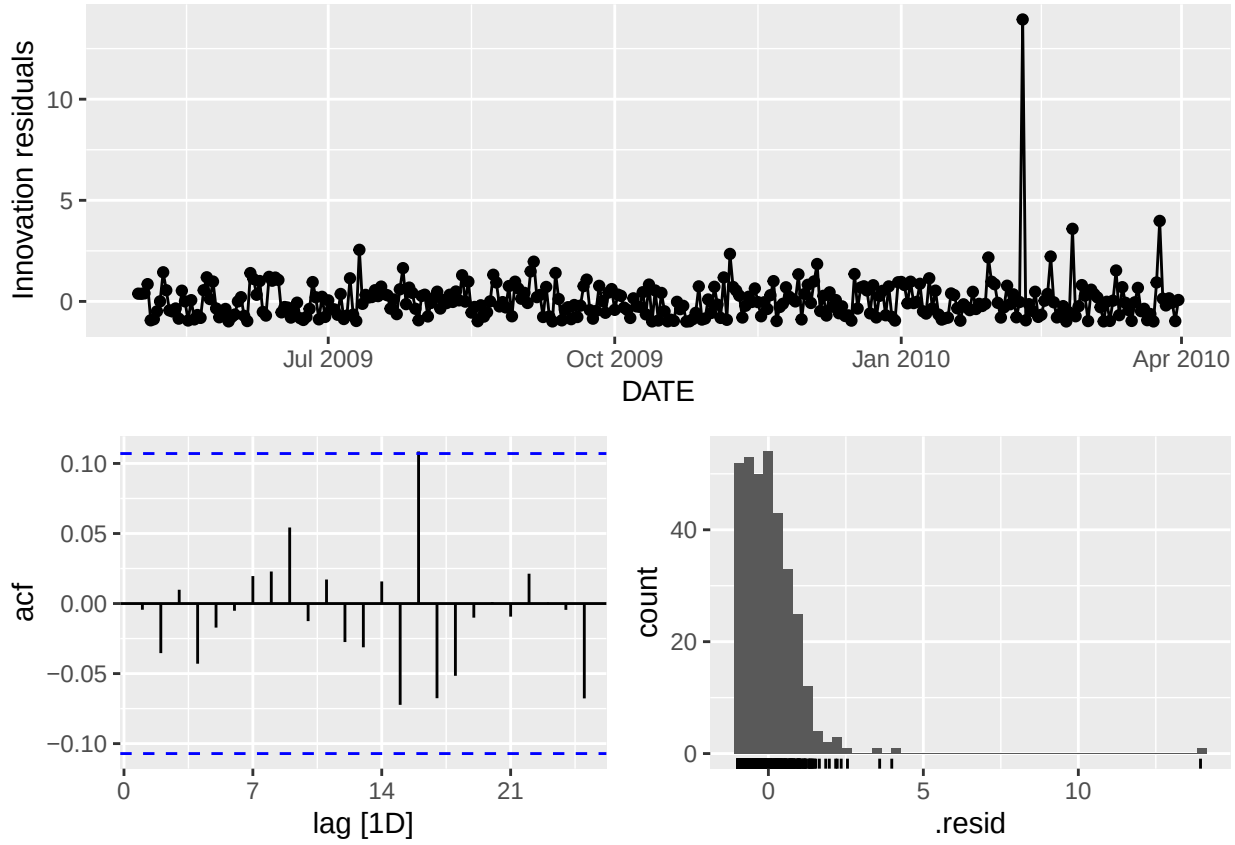

```
#splitting the data
train4 <- ATM_4 %>%
  filter(DATE <= as_date('2010-03-31'))

test4 <- ATM_4 %>%
  filter(DATE > as_date('2010-03-31'))
```

```
# fit for ETS model
ets_fit4 <- train4 %>%
  model(ETS(Cash))

# report ETS model
report(ets_fit4)
```

```
## Series: Cash
## Model: ETS(M,N,A)
##   Smoothing parameters:
##     alpha = 0.02672314
##     gamma = 0.0001000012
##
##   Initial states:
##      l[0]      s[0]     s[-1]   s[-2]    s[-3]    s[-4]     s[-5]    s[-6]
##  468.0227 -355.6519 -47.28137 307.745 5.989442 88.68834 -93.06094 93.57137
##
##   sigma^2:  1.21
```

```
##
##      AIC     AICc      BIC
## 6053.663 6054.342 6091.805
```

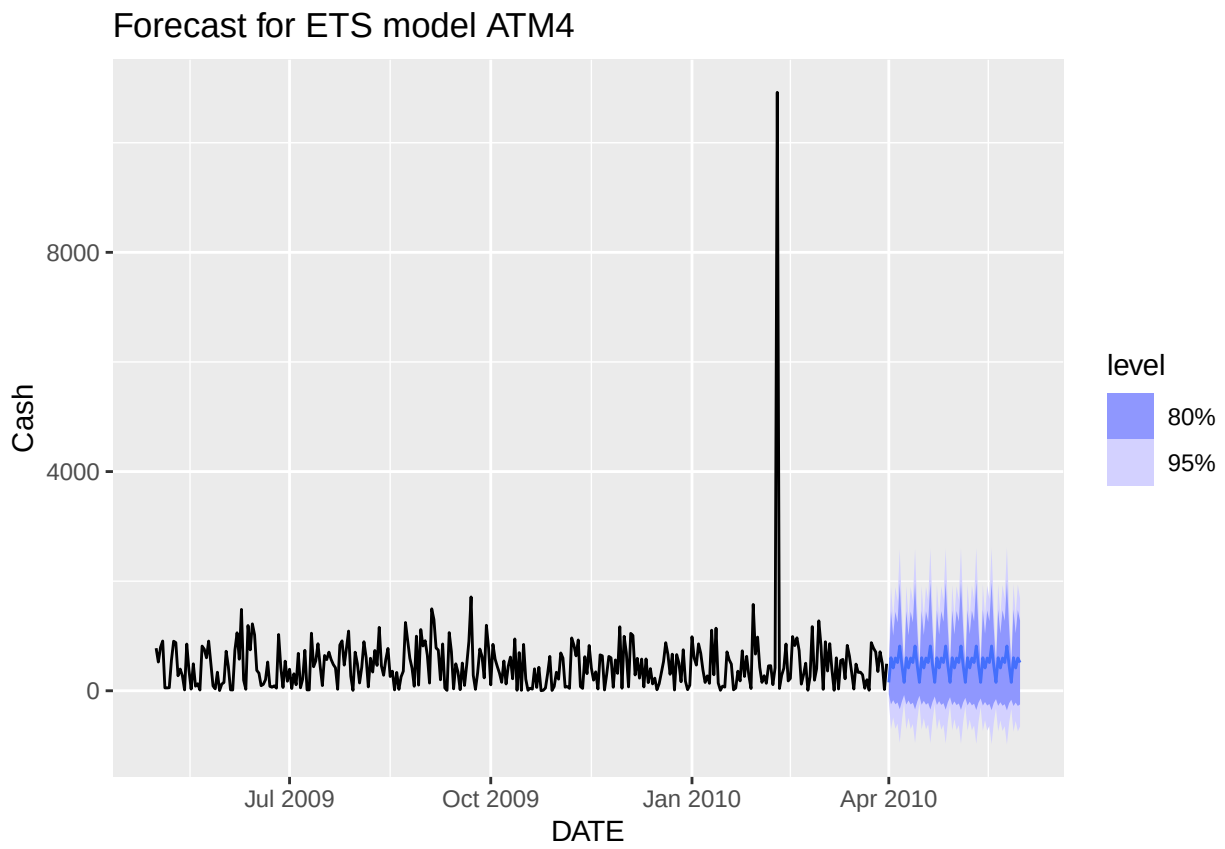
```
#residuals
gg_tsresiduals(ets_fit4)
```




```
#Ljung box test- White noise or not
ets_fit4 %>%
  augment() %>%
  features(.resid, ljung_box, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model    lb_stat lb_pvalue
##   <chr>       <dbl>     <dbl>
## 1 ETS(Cash)    9.58     0.996
```

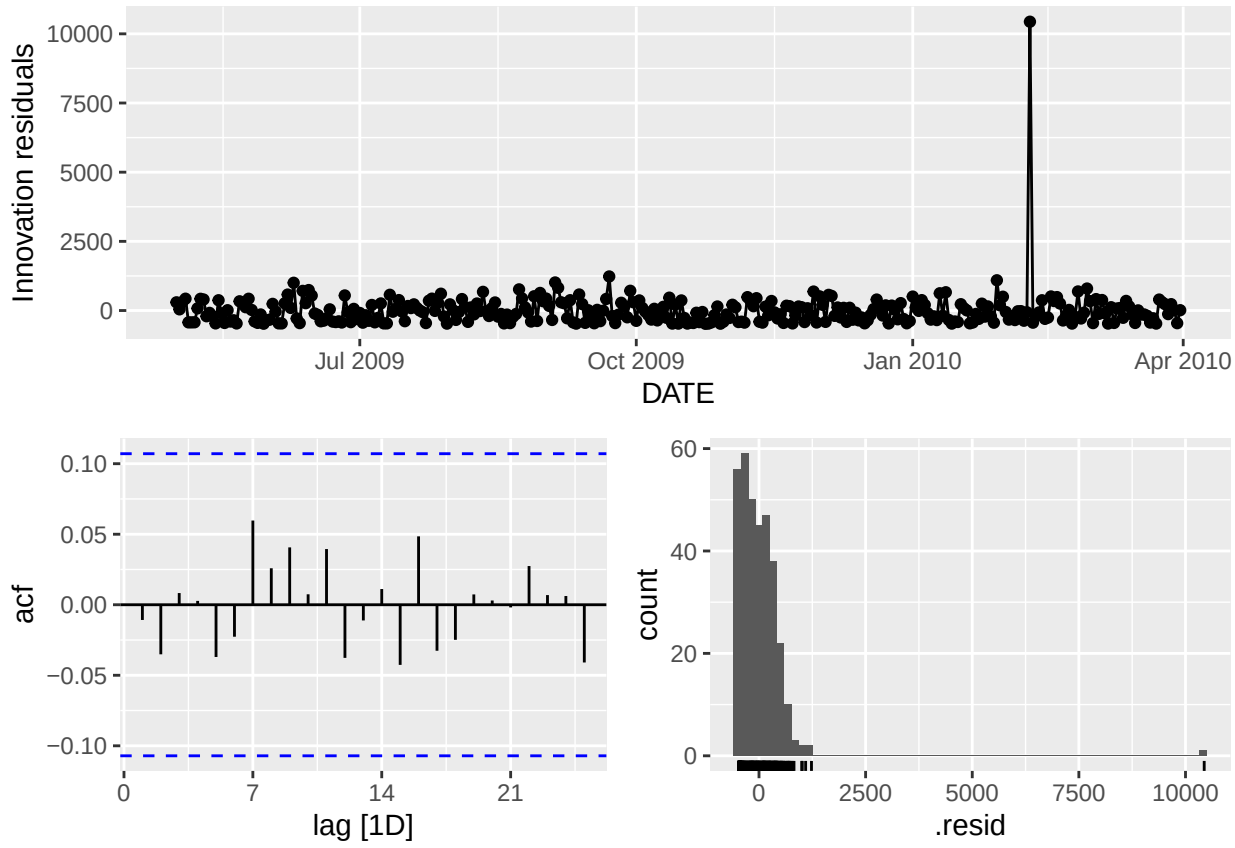
```
# ETS forecast
ets_fc4 <- ets_fit4 %>%
  forecast(h = '2 month')

#plot ets forecast
ets_fc4 %>%
  autoplot(train4) +
  labs(title = "Forecast for ETS model ATM4")
```




```
# fit for ARIMA model
arima_fit4 <- train4 %>%
  model(ARIMA(Cash))

#report on ARIMA model
report(arima_fit4)
```

```
## Series: Cash
## Model: ARIMA(0,0,0) w/ mean
##
## Coefficients:
##       constant
##       481.0987
## s.e.   36.7622
##
## sigma^2 estimated as 454093:  log likelihood=-2656.71
## AIC=5317.42   AICc=5317.45   BIC=5325.04
```

```
# residuals
gg_tsresiduals(arima_fit4)
```



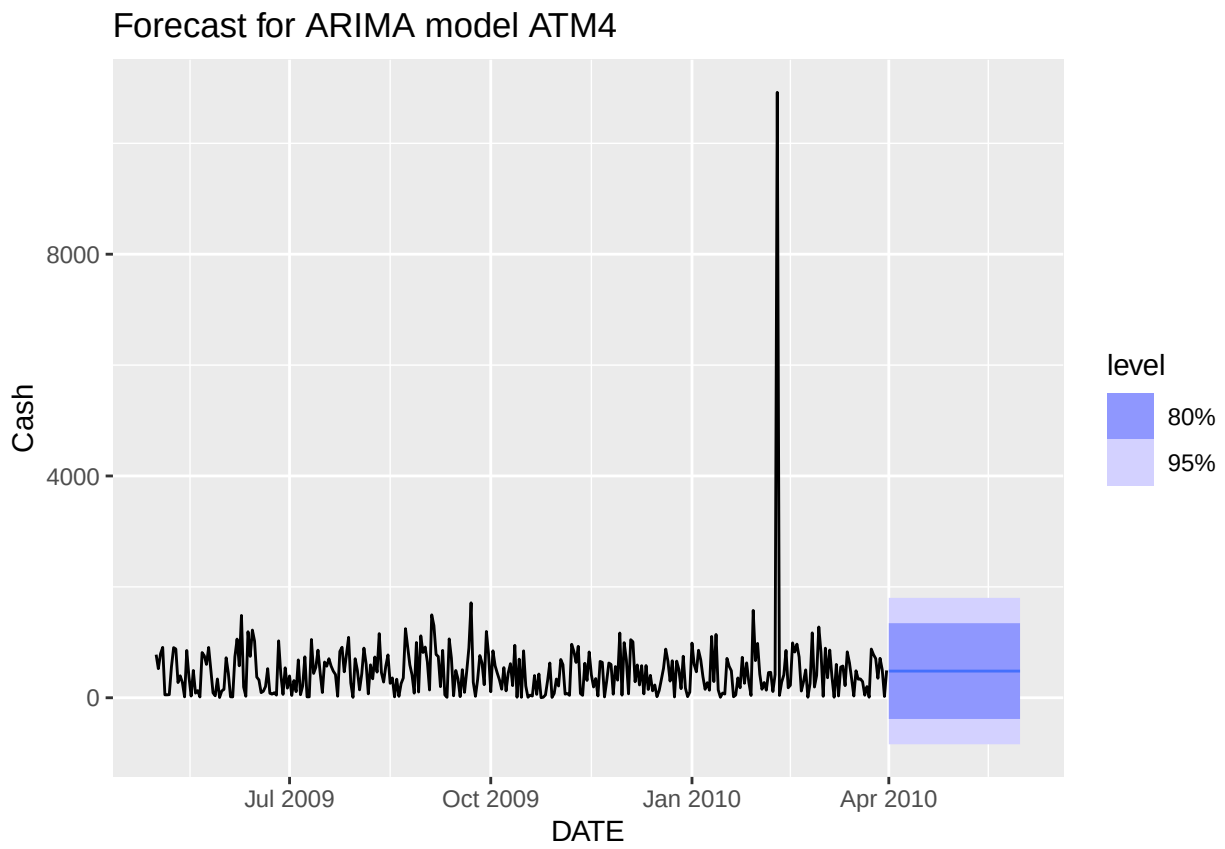

```
# Augmented Dickey-Fuller
adf_test4 <- adf.test(train4$Cash)
```

```
## Warning in adf.test(train4$Cash): p-value smaller than printed p-value
```

```
print(adf_test4)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train4$Cash
## Dickey-Fuller = -6.773, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
# ARIMA forecast
arima_fc4 <- arima_fit4 %>%
  forecast(h ='2 month')

#plot ARIMA forecast
arima_fc4 %>%
  autoplot(train4) +
  labs(title = "Forecast for ARIMA model ATM4")
```




```
# Accuracy of ARIMA forecast - MAE and RMSE
accuracy(arima_fc4, test4)
```

```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 31 observations are missing between 2010-05-01 and 2010-05-31
```

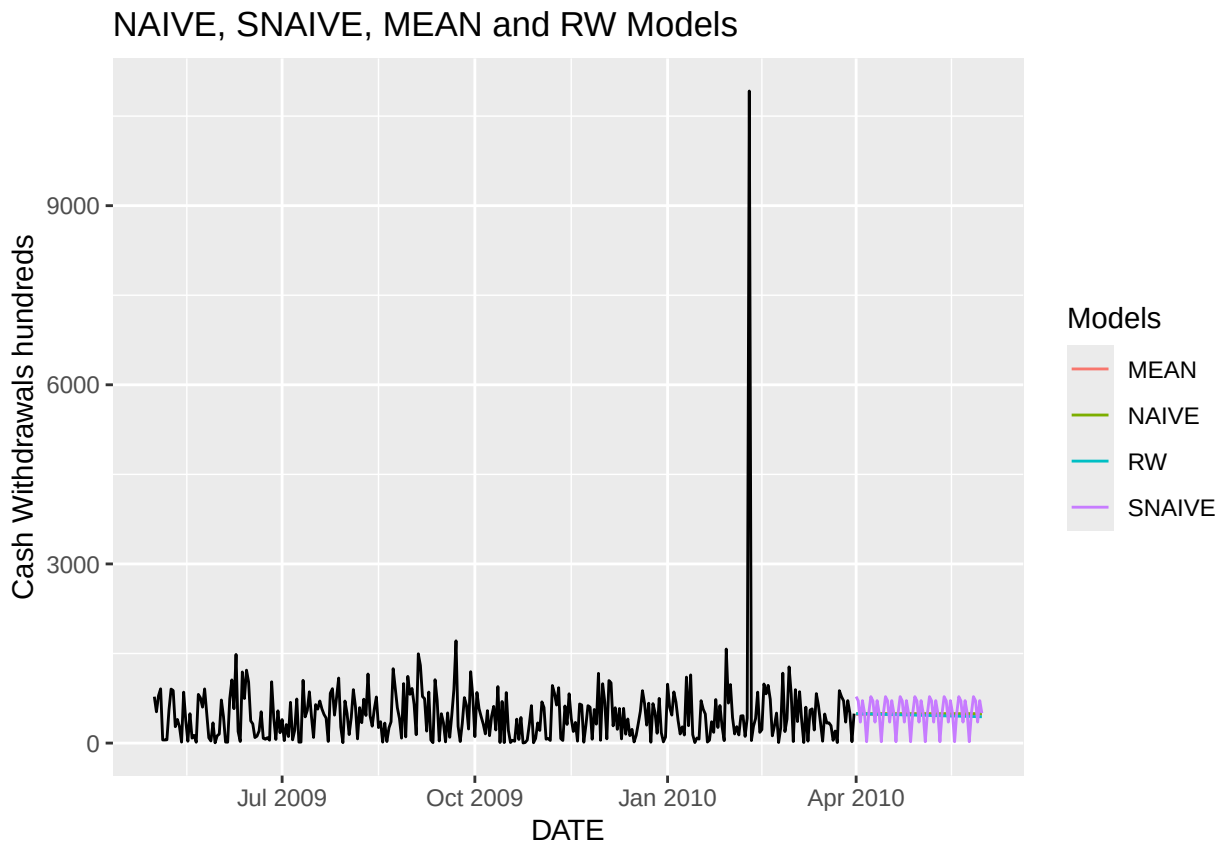
```
## # A tibble: 1 x 10
##   .model      .type    ME  RMSE   MAE    MPE  MAPE  MASE RMSSE     ACF1
##   <chr>       <chr> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1 ARIMA(Cash) Test  -85.8  293.  249. -1219. 1240.   NaN   NaN -0.00380
```

```
# Accuracy of ETS forecast - MAE and RMSE
accuracy(ets_fc4, test4)
```

```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 31 observations are missing between 2010-05-01 and 2010-05-31
```

```
## # A tibble: 1 x 10
##   .model    .type    ME  RMSE   MAE    MPE  MAPE  MASE RMSSE   ACF1
##   <chr>     <chr> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ETS(Cash) Test  -105.  386.  302. -1759. 1791.   NaN   NaN 0.0591
```

```
model_fit4 <- train4 %>%
  model(
    NAIVE = NAIVE(Cash),
    SNAIVE = SNAIVE(Cash),
    MEAN = MEAN(Cash),
    RW = RW(Cash ~ drift())
  )

model_fc4 <- model_fit4 %>%
  forecast(h = '2 month')

model_fc4 %>%
  autoplot(train4, level = NULL) +
  labs(title = "NAIVE, SNAIVE, MEAN and RW Models",  y = 'Cash Withdrawals hundreds') +
  guides(colour = guide_legend(title = "Models"))
```




```
#accuracy on model - lowest MAE, RMSE or MAPE
accuracy(model_fc4, test4)
```

```
## Warning: The future dataset is incomplete, incomplete out-of-sample data will be treated as missing.
## 31 observations are missing between 2010-05-01 and 2010-05-31
```

```
## # A tibble: 4 x 10
##   .model .type     ME  RMSE   MAE    MPE  MAPE  MASE RMSSE     ACF1
```

```
##   <chr>  <chr>  <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>    <dbl>
## 1 MEAN   Test   -85.8  293.  249. -1219. 1240.   NaN   NaN -0.00380
## 2 NAIVE  Test   -97.7  297.  253. -1251. 1271.   NaN   NaN -0.00380
## 3 RW     Test   -84.5  291.  248. -1218. 1240.   NaN   NaN -0.0168
## 4 SNAIVE Test  -130.   301.  227.  -360.  375.   NaN   NaN  0.0298
```

```
# Export forecast to Excel
fc4_data <- as.data.frame(model_fc4) %>%
  filter(.model == "SNAIVE")
write.xlsx(fc4_data, "Forecast_ATM4_FC.xlsx")
```

Part B

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.

```
# import data to R
power_usage <- read_excel("C:/Users/vitug/OneDrive/Desktop/CUNY Masters/DATA_624/ResidentialCustomerFor
head(power_usage)
```

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

```
str(power_usage)
```

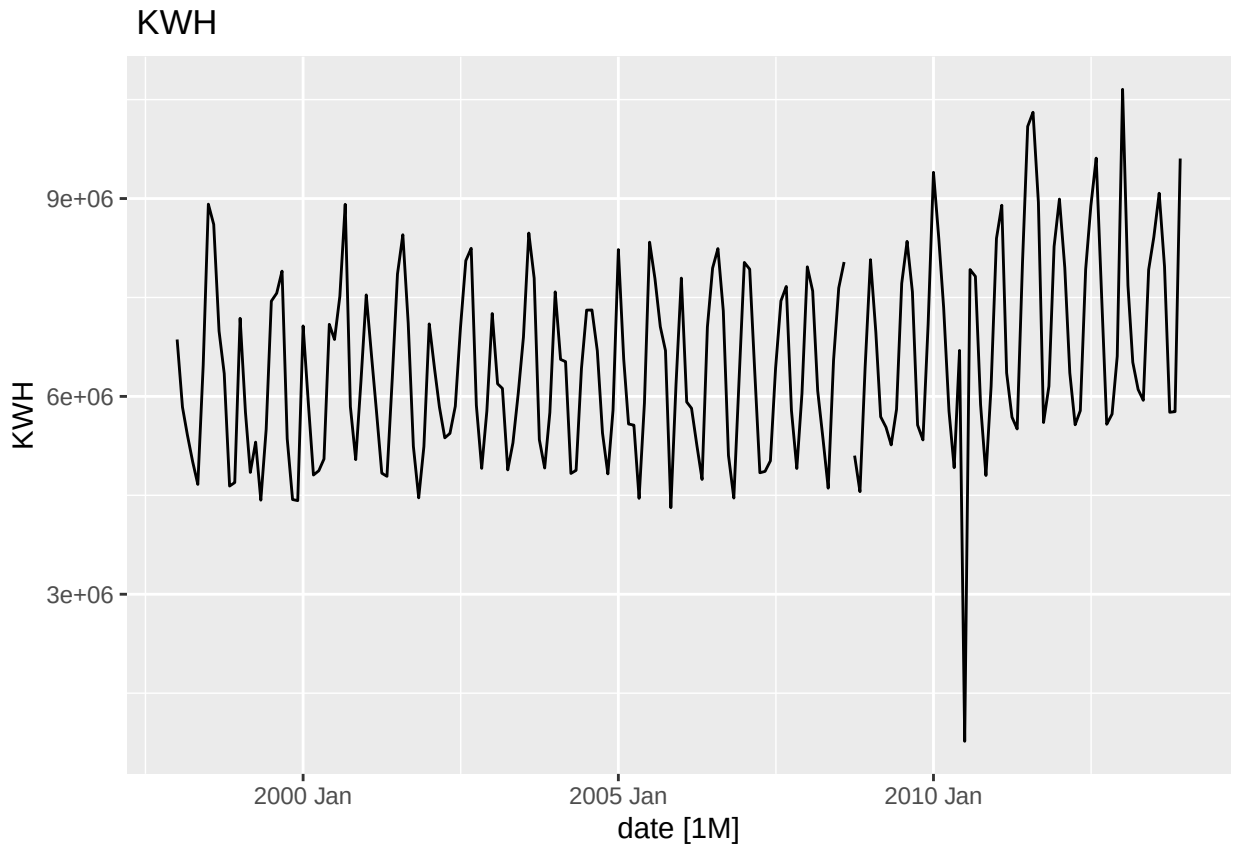
```
## tibble [192 x 3] (S3: tbl_df/tbl/data.frame)
##  $ CaseSequence: num [1:192] 733 734 735 736 737 738 739 740 741 742 ...
##  $ YYYY-MMM    : chr [1:192] "1998-Jan" "1998-Feb" "1998-Mar" "1998-Apr" ...
##  $ KWH         : num [1:192] 6862583 5838198 5420658 5010364 4665377 ...
```

```
#convert the date column from character to date
power_usage <- power_usage %>%
  mutate(date = ym(`YYYY-MMM`)) %>%
  select(-`YYYY-MMM`)

power_usage
```

```
## # A tibble: 192 x 3
##    CaseSequence     KWH date
##           <dbl>   <dbl> <date>
##  1          733 6862583 1998-01-01
##  2          734 5838198 1998-02-01
```

```
##  3          735 5420658 1998-03-01
##  4          736 5010364 1998-04-01
##  5          737 4665377 1998-05-01
##  6          738 6467147 1998-06-01
##  7          739 8914755 1998-07-01
##  8          740 8607428 1998-08-01
##  9          741 6989888 1998-09-01
## 10          742 6345620 1998-10-01
## # i 182 more rows
```

```r
# convert to tsibble
power <- power_usage %>%
  mutate(date = yearmonth(date)) %>%
  as_tsibble(index = date)

# plot the KWH column
power %>%
  autoplot(KWH) +
  labs(title = " KWH ")
```




```r
# summary data for missing values
summary(power_usage)
```

```
##    CaseSequence         KWH                 date
##  Min.   :733.0   Min.   :  770523   Min.   :1998-01-01
```

```
##  1st Qu.:780.8   1st Qu.: 5429912   1st Qu.:2001-12-24
##  Median :828.5   Median : 6283324   Median :2005-12-16
##  Mean   :828.5   Mean   : 6502475   Mean   :2005-12-15
##  3rd Qu.:876.2   3rd Qu.: 7620524   3rd Qu.:2009-12-08
##  Max.   :924.0   Max.   :10655730   Max.   :2013-12-01
##                  NA's   :1
```

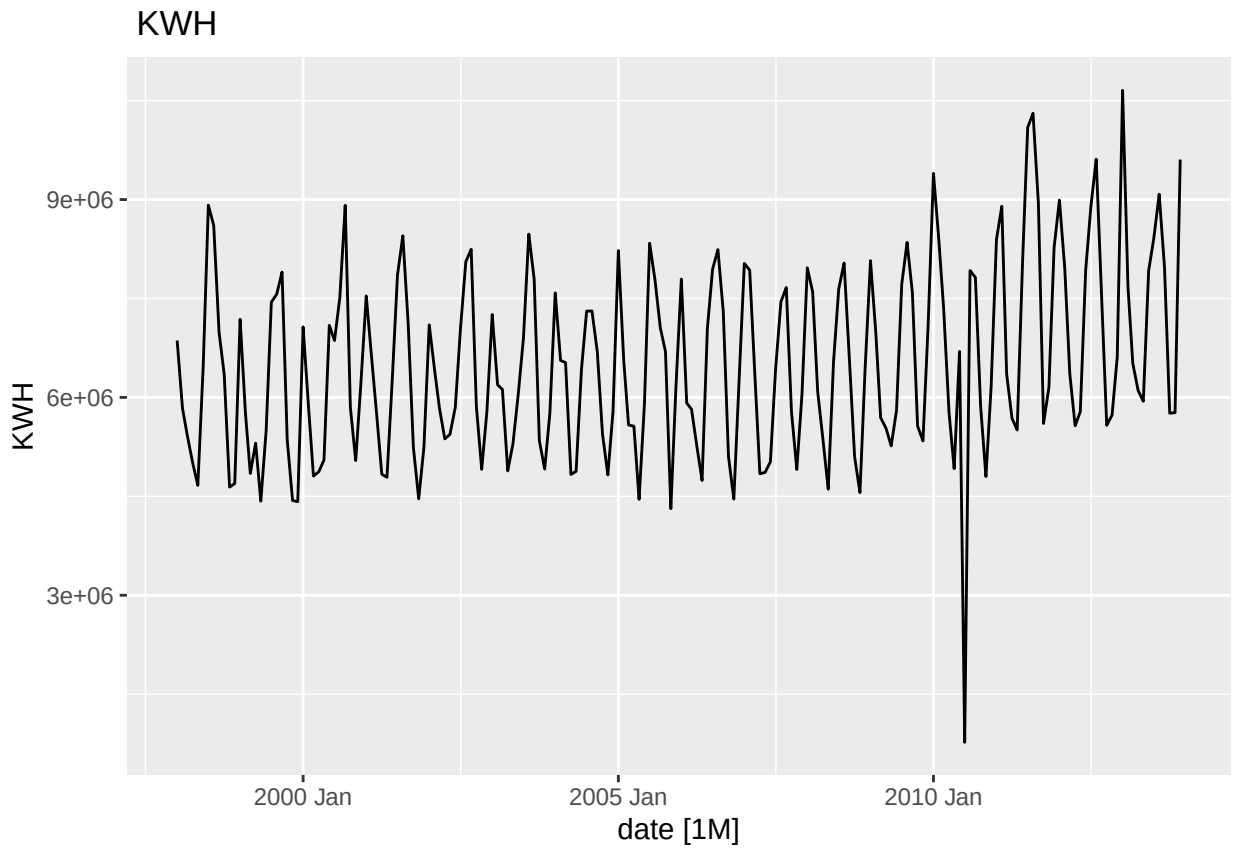
```
#view missing data
power %>%
  filter(is.na(KWH))
```

```
## # A tsibble: 1 x 3 [1M]
##   CaseSequence   KWH     date
##          <dbl> <dbl>    <mth>
## 1          861    NA 2008 Sep
```

```
# use na.approx to deal with the missing values
power2 <- power %>%
  mutate(KWH = na.approx(KWH))

# display column to make sure missing value it's been fulfilled
power2[129,]
```

```
## # A tsibble: 1 x 3 [1M]
##   CaseSequence     KWH     date
##          <dbl>   <dbl>    <mth>
## 1          861 6569470 2008 Sep
```

```
# plot the KWH column with new values
power2 %>%
  autoplot(KWH) +
  labs(title = " KWH ")
```




```
### Part B {.tabset}
```

Part B has less missing values on the dataset, there were only one column containing missing data, I used the same technique that I used in part one to fill missing values (na.appox). I split data into training and testing. I conducted a box-pierce test to see if the ETS model contained white noise on it before conducting forecasting. I applied the Arima model. I checked the data using SDF to see if the data was stationary,I used a box cox transformation on the data and differencing to forecast the data. I came up with the conclusion that the ETS forecasting is a better fit for this data set than the Arima model.

```
#splitting the data
train_power <- power2 %>%
  filter(date <= yearmonth('2012 Dec'))

test_power <- power2 %>%
  filter(date > yearmonth('2012 Dec'))
```

Split data for testing and training

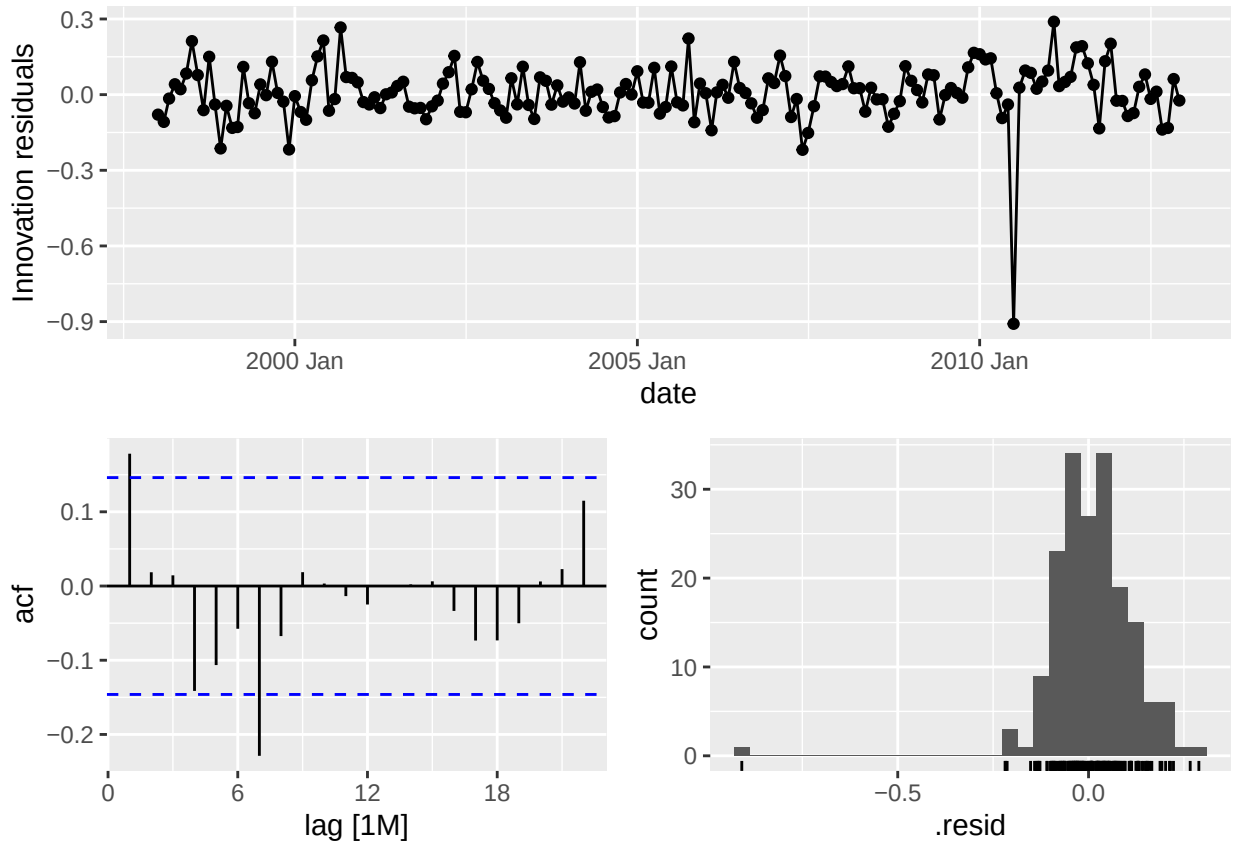
```
# fit ETS model
ets_powerfit <- train_power %>%
```

```
  model(ETS(KWH))

# report ETS model
report(ets_powerfit)
```

ETS Model

```
## Series: KWH
## Model: ETS(M,N,M)
##   Smoothing parameters:
##     alpha = 0.1459058
##     gamma = 0.0001000236
##
##   Initial states:
##     l[0]      s[0]     s[-1]     s[-2]    s[-3]   s[-4]   s[-5]     s[-6]
##  6136120 0.9341545 0.7517998 0.8776749 1.174301 1.27355 1.20892 0.9929466
##      s[-7]     s[-8]     s[-9]   s[-10]   s[-11]
##  0.7623283 0.8079936 0.9224428 1.079108 1.214781
##
##   sigma^2:  0.0139
##
##      AIC     AICc      BIC
## 5815.478 5818.405 5863.373
```

```
#residuals
gg_tsresiduals(ets_powerfit)
```



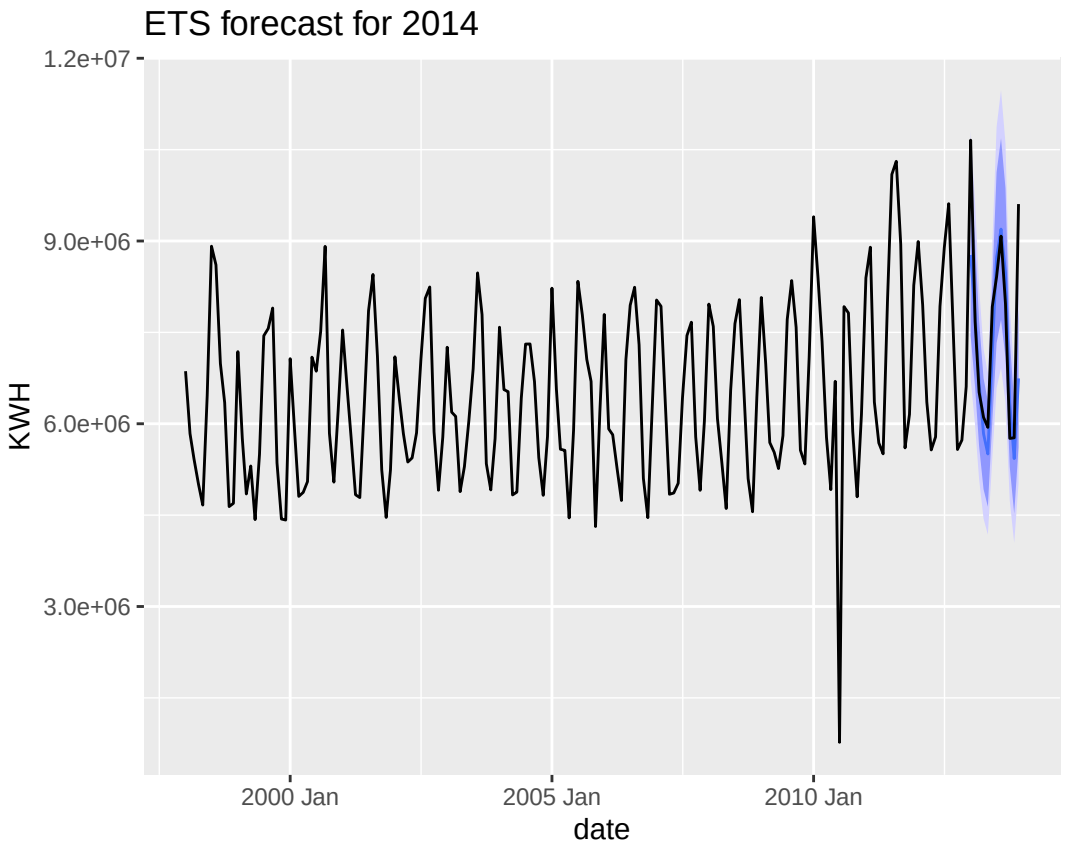
```r
#Ljung box test- White noise since p-value is over .05
ets_powerfit %>%
  augment() %>%
  features(.resid, ljung_box, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model   lb_stat lb_pvalue
##   <chr>      <dbl>     <dbl>
## 1 ETS(KWH)    27.1     0.302
```

```r
# Box Pierce test - no significant autocorrelation
ets_powerfit %>%
  augment() %>%
    features(.innov, box_pierce, lag = 24)
```

```
## # A tibble: 1 x 3
##   .model   bp_stat bp_pvalue
##   <chr>      <dbl>     <dbl>
## 1 ETS(KWH)    29.1     0.217
```

```r
#forecast ETS model
ets_powerfc <- ets_powerfit %>%
  forecast(h = '12 month')

# plot the forecast
ets_powerfc %>%
  autoplot(power2) +
  labs(title = "ETS forecast for 2014", y = 'KWH', x = 'date')
```



Forecast ETS Model

```
# Accuracy of ETS forecast - MAE and RMSE
accuracy(ets_powerfc, test_power)
```

```
## # A tibble: 1 x 10
##   .model   .type      ME     RMSE     MAE   MPE  MAPE  MASE RMSSE   ACF1
##   <chr>    <chr>   <dbl>    <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 ETS(KWH) Test  398169. 1057623. 692844.  4.14  8.31   NaN   NaN 0.0850
```

```
# Export forecast to Excel
fc5_data <- as.data.frame(ets_powerfc)
write.xlsx(fc5_data, "Forecast_Power_FC.xlsx")
```

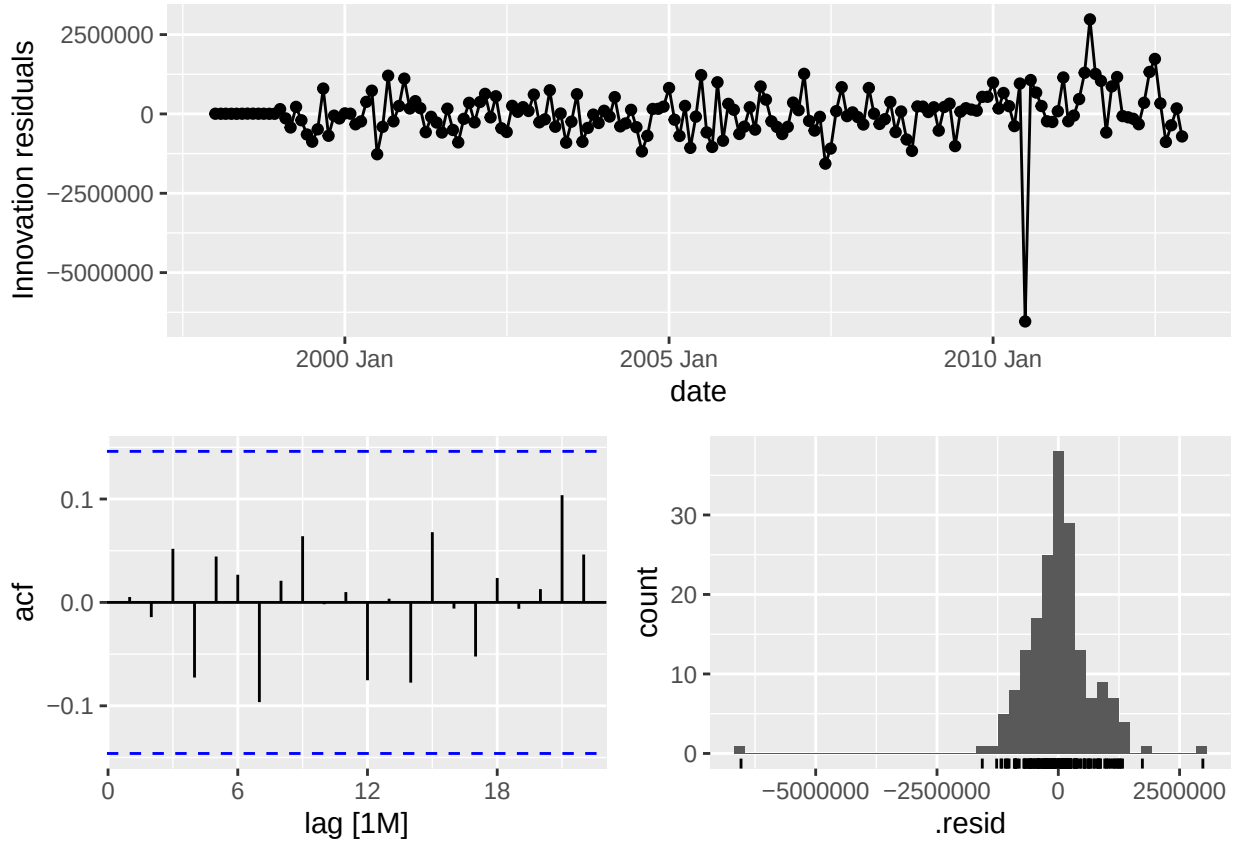
```
# fit for ARIMA model
arima_powerfit <- train_power %>%
  model(ARIMA(KWH))

#report on ARIMA model
report(arima_powerfit)
```

Arima Model

```
## Series: KWH
```

```
## Model: ARIMA(1,0,1)(2,1,0)[12] w/ drift
##
## Coefficients:
##          ar1      ma1     sar1     sar2  constant
##       0.6492  -0.4644  -0.8448  -0.6399  73967.36
## s.e.  0.2259   0.2632   0.0670   0.0756  35666.94
##
## sigma^2 estimated as 6.815e+11:  log likelihood=-2532.85
## AIC=5077.69   AICc=5078.22   BIC=5096.44
```

```
# residuals
gg_tsresiduals(arima_powerfit)
```




```
# Augmented Dickey-Fuller
adf_testpow <- adf.test(train_power$KWH)
```

```
## Warning in adf.test(train_power$KWH): p-value smaller than printed p-value
```

```
print(adf_testpow)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  train_power$KWH
## Dickey-Fuller = -4.8787, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```
#lambda
lambda <- power2 %>%
  features(KWH, features = guerrero) %>%
  pull(lambda_guerrero)

#box-cox transformation
power2 %>%
  features(box_cox(KWH,lambda), unitroot_ndiffs)
```

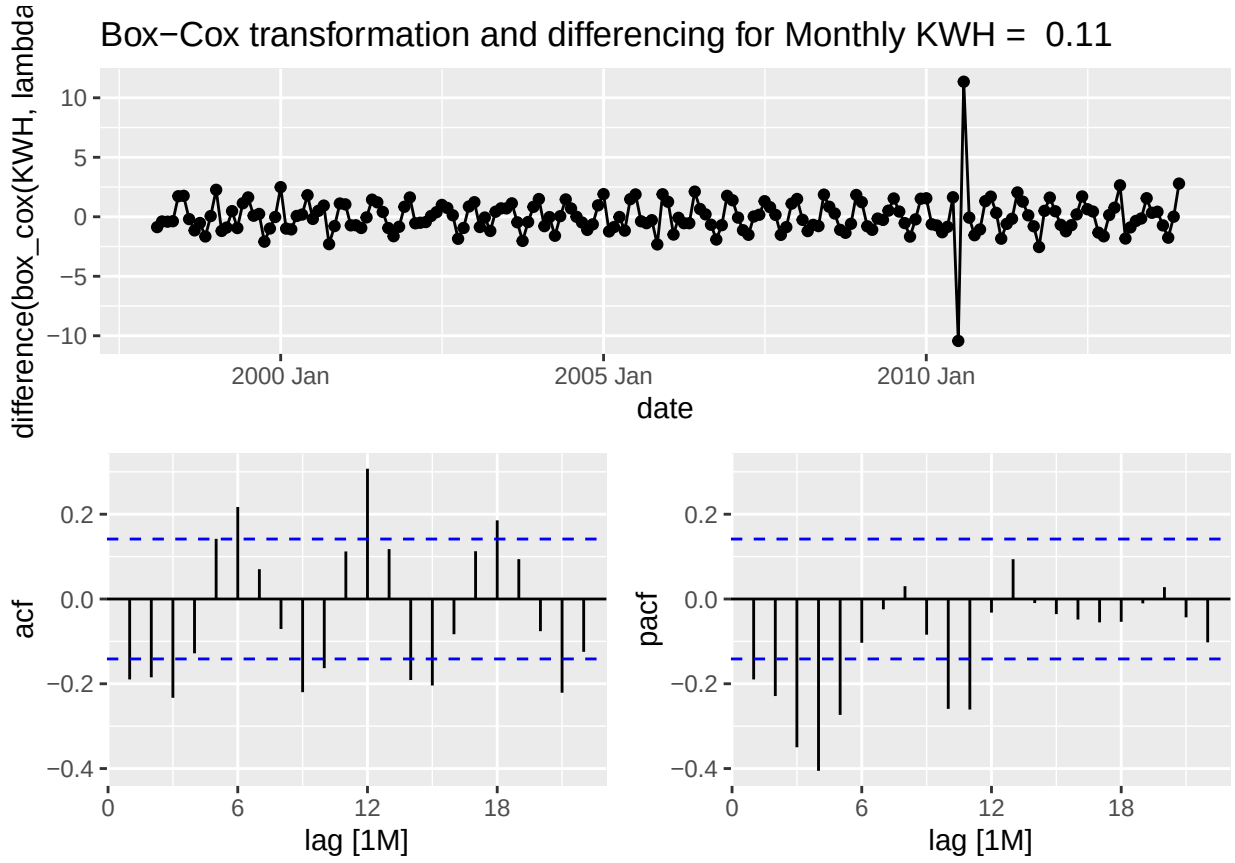
Lambda and Box-Cox transformation

```
## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      1
```

```
#display ACF and PACF with box_cox transformation and difference
power2 %>%
  gg_tsdisplay(difference(box_cox(KWH, lambda)), plot_type = 'partial') +
  labs(title = paste("Box-Cox transformation and differencing for Monthly KWH = ", round(lambda, 2)))
```

```
## 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()`).
```




```
#fit model
power_fit <- power2 %>%
  model (
    ARIMA(box_cox(KWH,lambda)),
    arima111 = ARIMA(box_cox(KWH,lambda) ~ pdq(1,1,1)),
   arima210 = ARIMA(box_cox(KWH,lambda) ~ pdq(2,1,0)),
   arima202 = ARIMA(box_cox(KWH,lambda) ~ pdq(2,0,2))
    )
```

```
## Warning: 1 error encountered for arima202
## [1] Could not find an appropriate ARIMA model.
## This is likely because automatic selection does not select models with characteristic roots that may
## For more details, refer to https://otexts.com/fpp3/arima-r.html#plotting-the-characteristic-roots
```

```
report(power_fit)
```

```
## Warning in report.mdl_df(power_fit): 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 x 8
##   .model                      sigma2 log_lik   AIC  AICc   BIC ar_roots ma_roots
##   <chr>                        <dbl>   <dbl> <dbl> <dbl> <dbl> <list>   <list>
## 1 ARIMA(box_cox(KWH, lambda))   1.43   -305.  620.  620.  636. <cpl>    <cpl>
## 2 arima111                      1.44   -306.  621.  622.  638. <cpl>    <cpl>
## 3 arima210                      1.98   -335.  680.  681.  697. <cpl>    <cpl>
```

```r
glance(power_fit) %>% arrange(AICc)
```

```
## # A tibble: 3 x 8
##   .model                       sigma2 log_lik   AIC  AICc   BIC ar_roots ma_roots
##   <chr>                         <dbl>   <dbl> <dbl> <dbl> <dbl> <list>   <list>
## 1 ARIMA(box_cox(KWH, lambda))   1.43   -305.  620.  620.  636. <cpl>    <cpl>
## 2 arima111                      1.44   -306.  621.  622.  638. <cpl>    <cpl>
## 3 arima210                      1.98   -335.  680.  681.  697. <cpl>    <cpl>
```

```r
#forecast
arima_fcpow <-  power2 %>%
  model (ARIMA(box_cox(KWH,lambda)))%>%
  forecast(h = "12 months")

# Plot the forecast
arima_fcpow %>%
  autoplot(power2) +
  labs(title = "Forecast for ARIMA Model for KWH")
```