Data624 - Project1

Amit Kapoor

3/28/2021

Contents

Overview

This project includes 3 time series dataset and requires to select best forecasting model for all 3 datasets.

- Part A - ATM Forecast
- Part B - Forecasting Power
- Part C - Waterflow Pipe

Part A - ATM Forecast

The dataset contains cash withdrawals from 4 different ATM machines from May 2009 to Apr 2010. The variable 'Cash' is provided in hundreds of dollars and data is in a single file. Before starting our analysis we will first download the excel from github and then read it through read_excel.

Exploratory Analysis

```
temp.file <- tempfile(fileext = ".xlsx")
download.file(url="https://github.com/amit-kapoor/data624/blob/main/Project1/ATM624Data.xlsx?raw=true",
              destfile = temp.file,
              mode = "wb",
              quiet = TRUE)
atm.data <- read_excel(temp.file, skip=0, col_types = c("date","text","numeric"))

glimpse(atm.data)
```

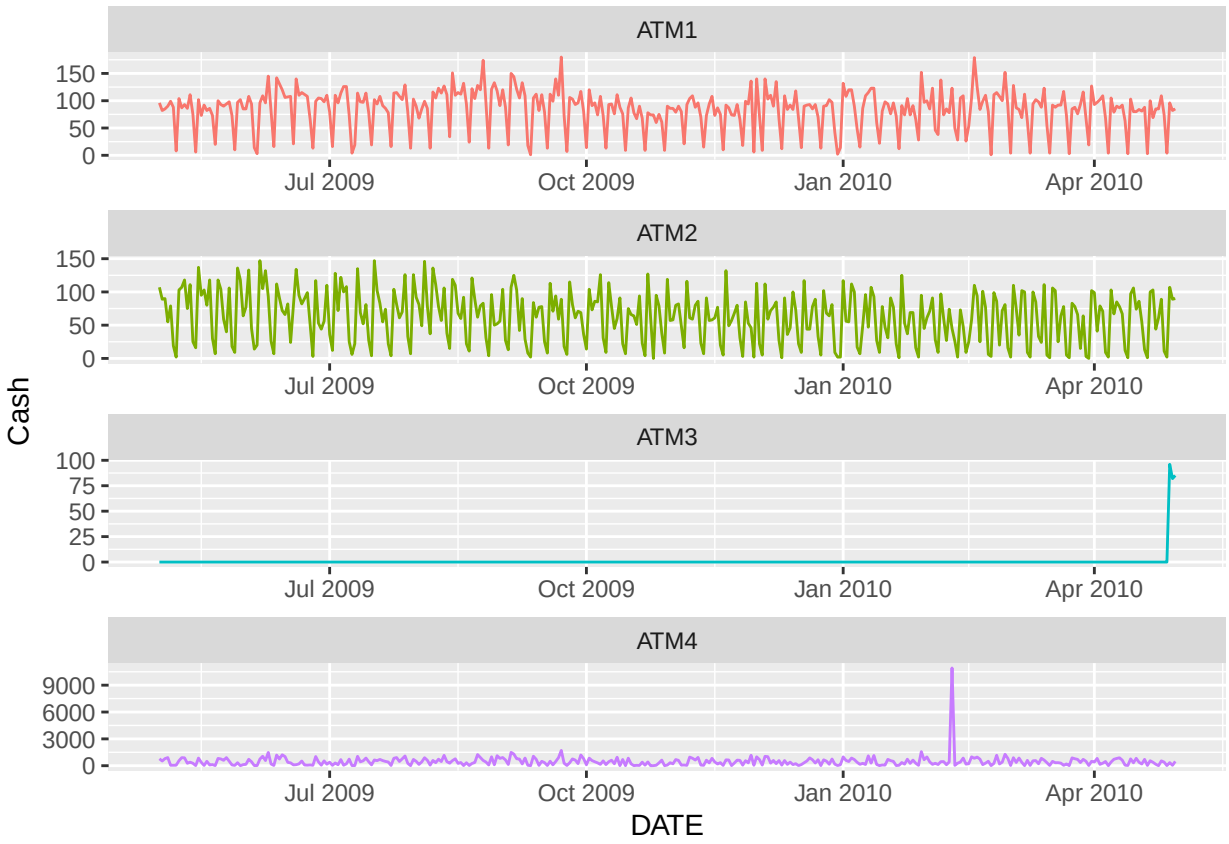
```
## Rows: 1,474
## Columns: 3
## $ DATE <dttm> 2009-05-01, 2009-05-01, 2009-05-02, 2009-05-02, 2009-05-03, 2009~
## $ ATM  <chr> "ATM1", "ATM2", "ATM1", "ATM2", "ATM1", "ATM2", "ATM1", "ATM2", "~
```

```
## $ Cash <dbl> 96, 107, 82, 89, 85, 90, 90, 55, 99, 79, 88, 19, 8, 2, 104, 103, ~
```

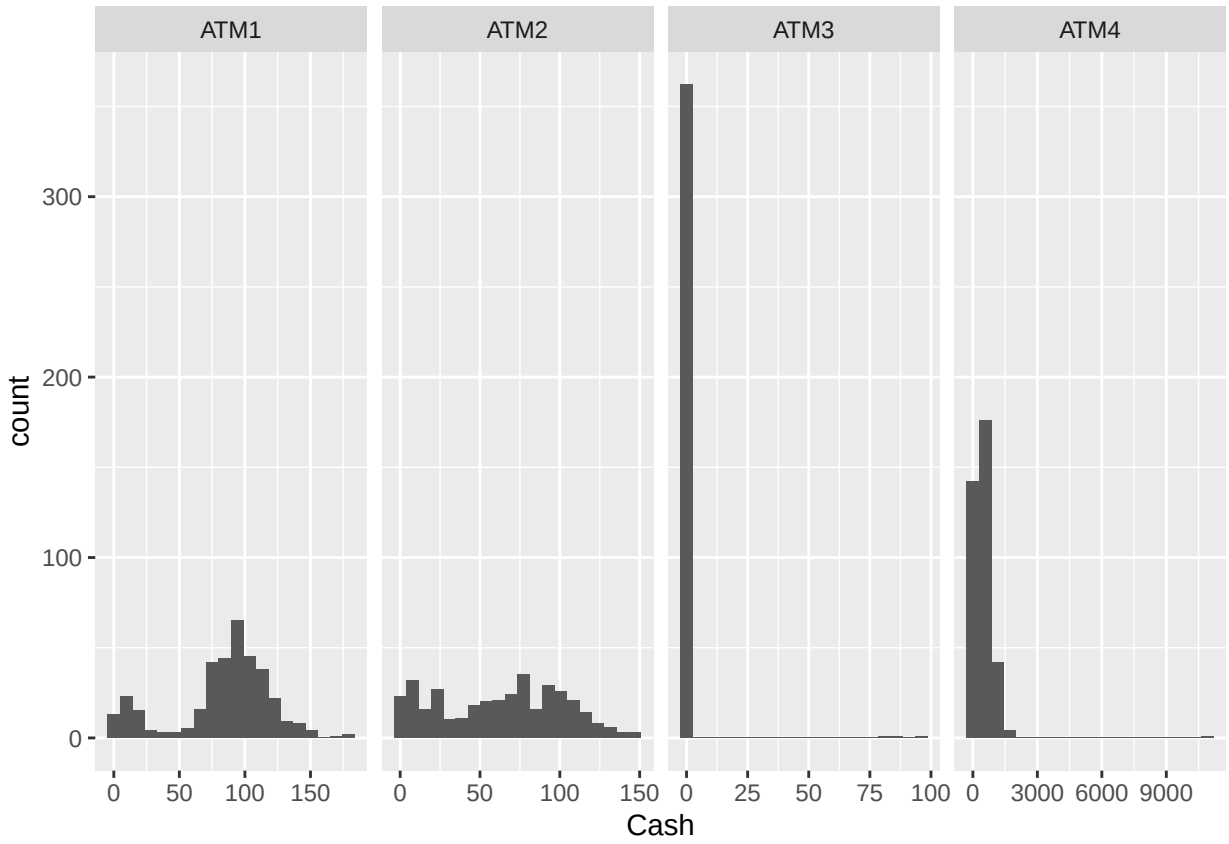
```
# rows missing values
atm.data[!complete.cases(atm.data),]
```

```
## # A tibble: 19 x 3
##    DATE                ATM    Cash
##    <dttm>              <chr> <dbl>
##  1 2009-06-13 00:00:00 ATM1     NA
##  2 2009-06-16 00:00:00 ATM1     NA
##  3 2009-06-18 00:00:00 ATM2     NA
##  4 2009-06-22 00:00:00 ATM1     NA
##  5 2009-06-24 00:00:00 ATM2     NA
##  6 2010-05-01 00:00:00 <NA>     NA
##  7 2010-05-02 00:00:00 <NA>     NA
##  8 2010-05-03 00:00:00 <NA>     NA
##  9 2010-05-04 00:00:00 <NA>     NA
## 10 2010-05-05 00:00:00 <NA>     NA
## 11 2010-05-06 00:00:00 <NA>     NA
## 12 2010-05-07 00:00:00 <NA>     NA
## 13 2010-05-08 00:00:00 <NA>     NA
## 14 2010-05-09 00:00:00 <NA>     NA
## 15 2010-05-10 00:00:00 <NA>     NA
## 16 2010-05-11 00:00:00 <NA>     NA
## 17 2010-05-12 00:00:00 <NA>     NA
## 18 2010-05-13 00:00:00 <NA>     NA
## 19 2010-05-14 00:00:00 <NA>     NA
```

```
ggplot(atm.data[complete.cases(atm.data),] , aes(x=DATE, y=Cash, col=ATM )) +
  geom_line(show.legend = FALSE) +
  facet_wrap(~ATM, ncol=1, scales = "free")
```




```
ggplot(atm.data[complete.cases(atm.data),] , aes(x=Cash )) +
  geom_histogram(bins=20) +
  facet_grid(cols=vars(ATM), scales = "free")
```




```
# consider complete cases
atm.comp <- atm.data[complete.cases(atm.data),]
# pivot wider with cols from 4 ATMs and their values as Cash
atm.comp <- atm.comp %>% pivot_wider(names_from = ATM, values_from = Cash)
head(atm.comp)
```

```
## # A tibble: 6 x 5
##   DATE                ATM1  ATM2  ATM3  ATM4
##   <dttm>             <dbl> <dbl> <dbl> <dbl>
## 1 2009-05-01 00:00:00    96   107     0 777.
## 2 2009-05-02 00:00:00    82    89     0 524.
## 3 2009-05-03 00:00:00    85    90     0 793.
## 4 2009-05-04 00:00:00    90    55     0 908.
## 5 2009-05-05 00:00:00    99    79     0  52.8
## 6 2009-05-06 00:00:00    88    19     0  52.2
```

```
# summary
atm.comp %>% select(-DATE) %>% summary()
```

```
##       ATM1             ATM2             ATM3              ATM4
##  Min.   :  1.00   Min.   :  0.00   Min.   : 0.0000   Min.   :    1.563
##  1st Qu.: 73.00   1st Qu.: 25.50   1st Qu.: 0.0000   1st Qu.:  124.334
##  Median : 91.00   Median : 67.00   Median : 0.0000   Median :  403.839
##  Mean   : 83.89   Mean   : 62.58   Mean   : 0.7206   Mean   :  474.043
##  3rd Qu.:108.00   3rd Qu.: 93.00   3rd Qu.: 0.0000   3rd Qu.:  704.507
##  Max.   :180.00   Max.   :147.00   Max.   :96.0000   Max.   :10919.762
##  NA's   :3        NA's   :2
```

Per above exploratory analysis, all ATMs show different patterns. We would perform forecasting for each

ATM separately.

- ATM1 and ATM2 shows similar pattern (approx.) throughout the time. ATM1 and ATM2 have 3 and 2 missing entries respectively.
- ATM3 appears to become online in last 3 days only and rest of days appears inactive. So tha data available for this ATM is very limited.
- ATM4 requires replacement for outlier and we can assume that one day spike of cash withdrawal is unique. It has an outlier showing withdrawl amount 10920.

Data Cleaning

For this part we will first apply ts() function to get required time series. Next step is to apply tsclean function that will handle missing data along with outliers. To estimate missing values and outlier replacements, this function uses linear interpolation on the (possibly seasonally adjusted) series. Once we get the clean data we will use pivot_longer to get the dataframe in its original form.

```
atm.ts <- ts(atm.comp %>% select(-DATE))
head(atm.ts)
```

```
## Time Series:
## Start = 1
## End = 6
## Frequency = 1
##   ATM1 ATM2 ATM3      ATM4
## 1   96  107    0 776.99342
## 2   82   89    0 524.41796
## 3   85   90    0 792.81136
## 4   90   55    0 908.23846
## 5   99   79    0  52.83210
## 6   88   19    0  52.20845
```

```
# apply tsclean
atm.ts.cln <- sapply(X=atm.ts, tsclean)
atm.ts.cln %>% summary()
```

```
##       ATM1             ATM2             ATM3             ATM4
##  Min.   :  1.00   Min.   :  0.00   Min.   : 0.0000   Min.   :   1.563
##  1st Qu.: 73.00   1st Qu.: 26.00   1st Qu.: 0.0000   1st Qu.: 124.334
##  Median : 91.00   Median : 67.00   Median : 0.0000   Median : 402.770
##  Mean   : 84.15   Mean   : 62.59   Mean   : 0.7206   Mean   : 444.757
##  3rd Qu.:108.00   3rd Qu.: 93.00   3rd Qu.: 0.0000   3rd Qu.: 704.192
##  Max.   :180.00   Max.   :147.00   Max.   :96.0000   Max.   :1712.075
```

If we compare this summary with previous one of original data, ATM1 and ATM2 has nomore NAs and ATM4 outlier value (10919.762) is handled and now the max value is 1712.075.

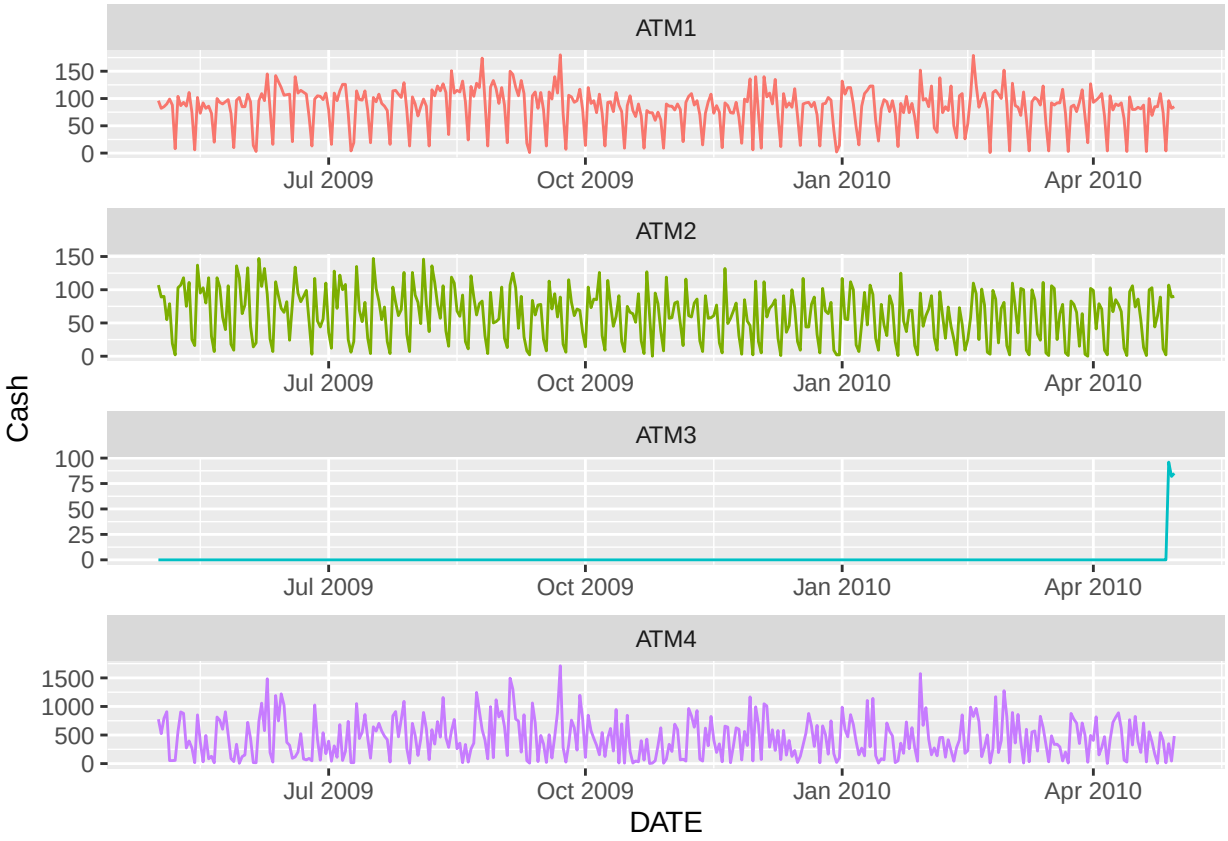
```
# convert into data frame, pivot longer , arrange by ATM and bind with dates
atm.new <- as.data.frame(atm.ts.cln) %>%
  pivot_longer(everything(), names_to = "ATM", values_to = "Cash") %>%
  arrange(ATM)

atm.new <- cbind(DATE = seq(as.Date("2009-05-1"), as.Date("2010-04-30"), length.out=365),
                 atm.new)

head(atm.new)
```

```
##         DATE  ATM Cash
## 1 2009-05-01 ATM1   96
```

```
## 2 2009-05-02 ATM1   82
## 3 2009-05-03 ATM1   85
## 4 2009-05-04 ATM1   90
## 5 2009-05-05 ATM1   99
## 6 2009-05-06 ATM1   88
```

```
#library(xlsx)
#write.xlsx(atm.new, 'atmnew.xlsx', sheetName = "Sheet1", col.names = TRUE, row.names = TRUE, append =
```

```
ggplot(atm.new , aes(x=DATE, y=Cash, col=ATM )) +
  geom_line(show.legend = FALSE) +
  facet_wrap(~ATM, ncol=1, scales = "free")
```



Though above plot doesn't show much differences for ATM1,2,3 but tsclean handled the ATM4 data very well after replacing the outlier.

Time Series

Function to plot forecast for various models.

```
# function to plot forecast(s)
atm.forecast <- function(timeseries) {
  # lambda value
  lambda <- BoxCox.lambda(timeseries)
  # models for forecast
  hw.model <- timeseries %>% hw(h=31, seasonal = "additive", lambda = lambda, damped = TRUE)
  ets.model <- timeseries %>% ets(lambda = lambda)
  arima.model <- timeseries %>% auto.arima(lambda = lambda)
  # forecast
```

```
  atm.hw.fcst <- forecast(hw.model, h=31)
  atm.ets.fcst <- forecast(ets.model, h=31)
  atm.arima.fcst <- forecast(arima.model, h=31)
  # plot forecasts
  p1 <- autoplot(timeseries) +
    autolayer(atm.hw.fcst, PI=FALSE, series="Holt-Winters") +
    autolayer(atm.ets.fcst, PI=FALSE, series="ETS") +
    autolayer(atm.arima.fcst, PI=FALSE, series="ARIMA") +
    theme(legend.position = "top") +
    ylab("Cash Withdrawl")
  # zoom in plot
  p2 <- p1 +
    labs(title = "Zoom in ") +
    xlim(c(51,56))

  grid.arrange(p1,p2,ncol=1)

}
```

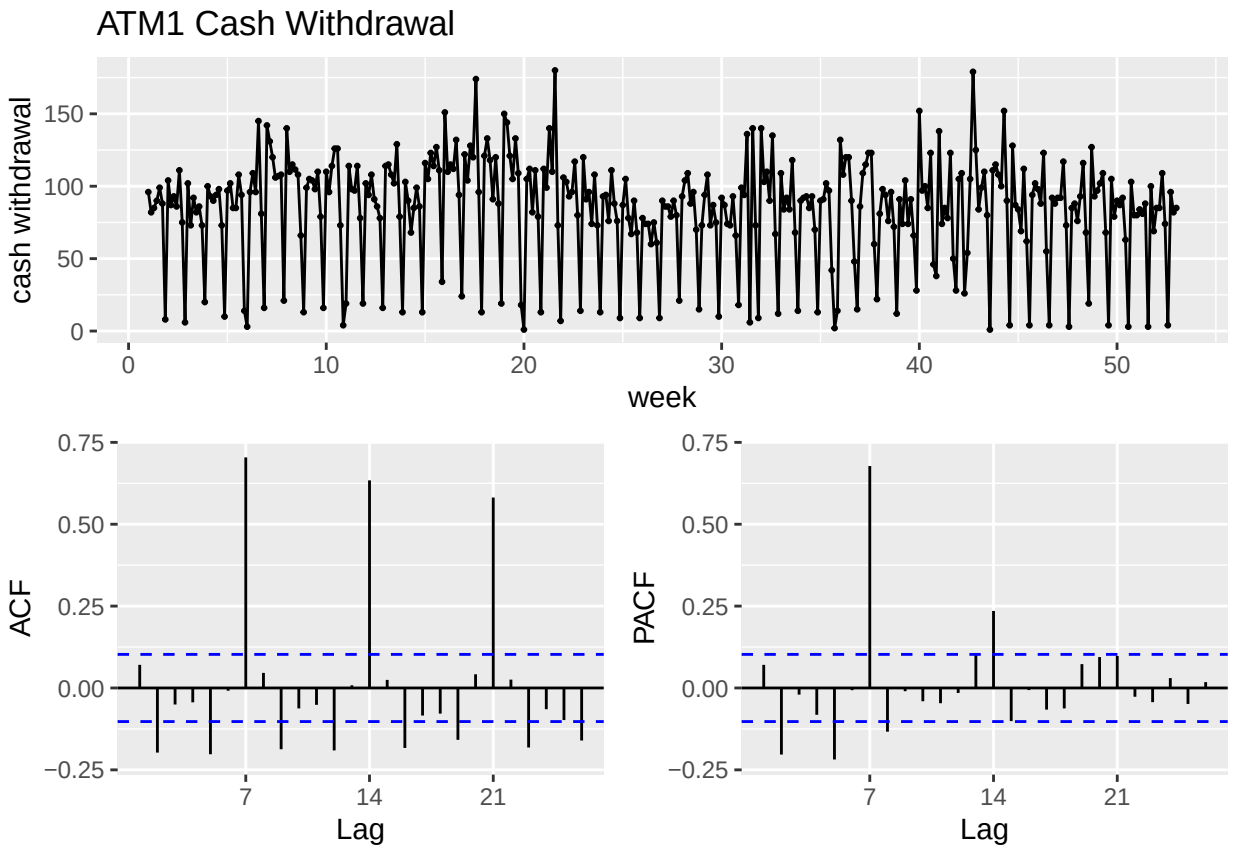
Function to calculate RMSEs for various models.

```
model_accuracy <- function(timeseries) {
  # lambda value
  lambda <- BoxCox.lambda(timeseries)
  # models for forecast
  hw.model <- timeseries %>% hw(h=31, seasonal = "additive", lambda = lambda, damped = TRUE)
  ets.model <- timeseries %>% ets(lambda = lambda)
  arima.model <- timeseries %>% auto.arima(lambda = lambda)
  # dataframe having rmse
  rmse = data.frame(RMSE=cbind(accuracy(hw.model)[,2],
                               accuracy(ets.model)[,2],
                               accuracy(arima.model)[,2]))
  names(rmse) = c("Holt-Winters", "ETS", "ARIMA")
  # display rmse
  rmse
}
```

ATM1

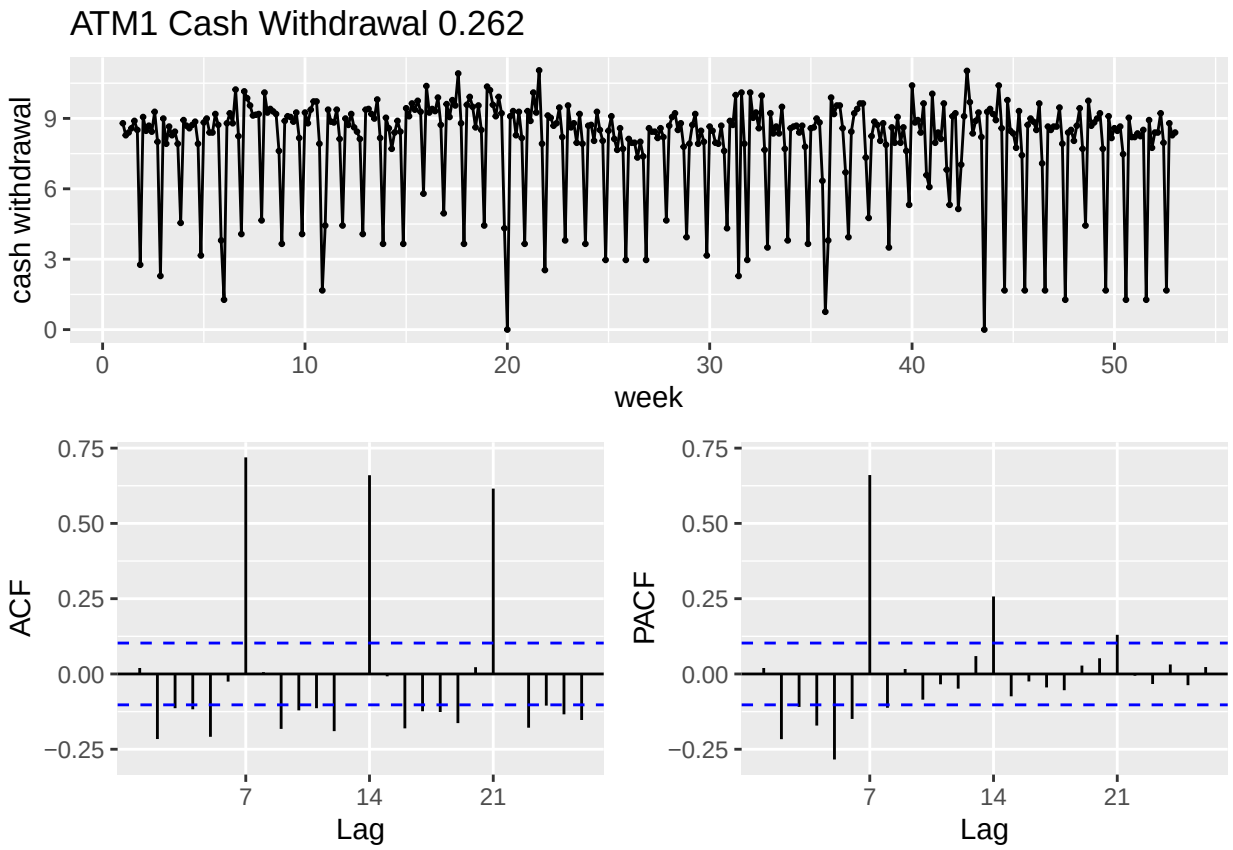
Seeing the time series plot, it is clear that there is a seasonality in the data. We can see increasing and decreasing activities over the weeks in below plot. From the ACF plot, we can see a slight decrease in every 7th lag due to trend. PACF plot shows some significant lags at the beginning.

```
atm1.ts <- atm.new %>% filter(ATM=="ATM1") %>% select(Cash) %>% ts(frequency = 7)
ggtsdisplay(atm1.ts, main="ATM1 Cash Withdrawal", ylab="cash withdrawal", xlab="week")
```



From the above plots it is evident that the time series is non stationary, showing seasonality and will require differencing to make it stationary.

```
atm1.lambda <- BoxCox.lambda(atm1.ts)
atm1.ts.bc <- BoxCox(atm1.ts, atm1.lambda )
ggtsdisplay(atm1.ts.bc, main=paste("ATM1 Cash Withdrawal",round(atm1.lambda, 3)), ylab="cash withdrawal
```




```
# Number of differences required for a stationary series
ndiffs(atm1.ts.bc)
```

```
## [1] 0
```

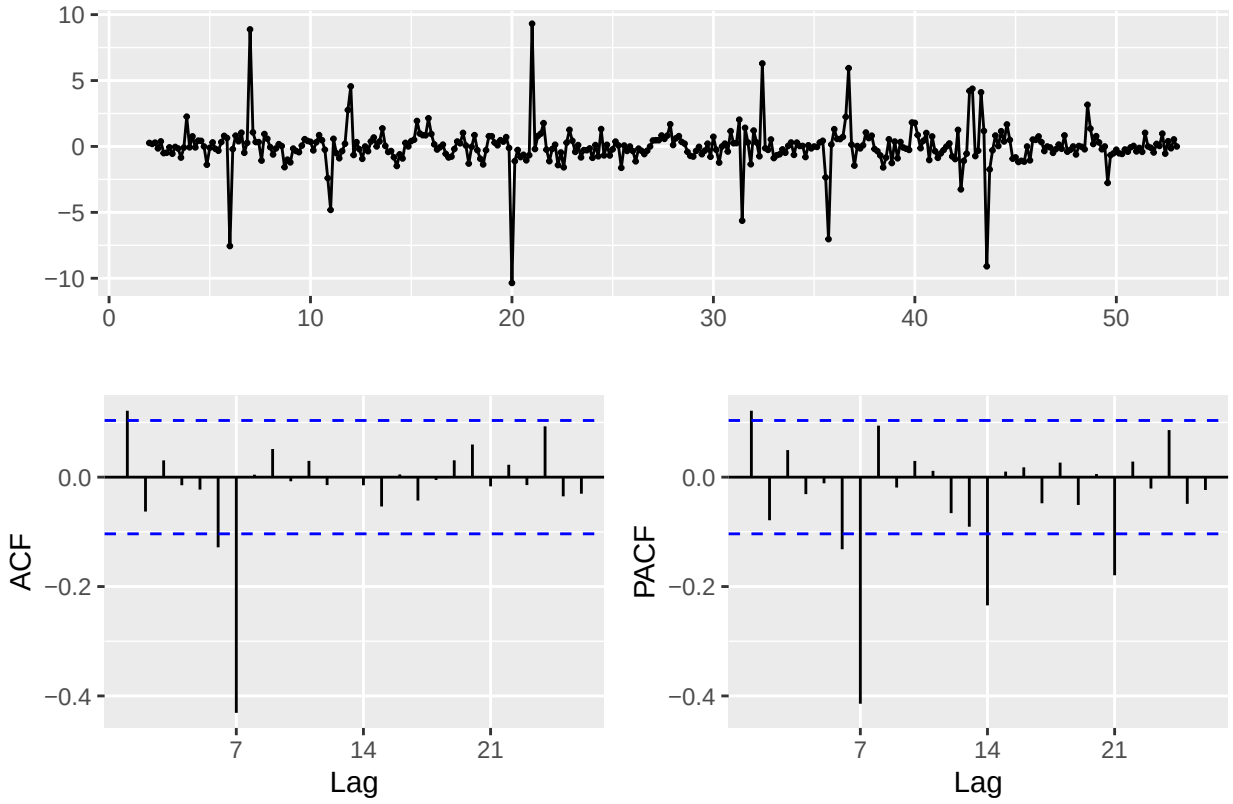
```
# Number of differences required for a seasonally stationary series
nsdiffs(atm1.ts.bc)
```

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

```
atm1.ts.bc %>% diff(lag=7) %>% ur.kpss() %>% summary()
```

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.0153
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
atm1.ts.bc %>% diff(lag=7) %>% ggtsdisplay()
```




```
atm1.ts %>% ets(lambda = atm1.lambda )
```

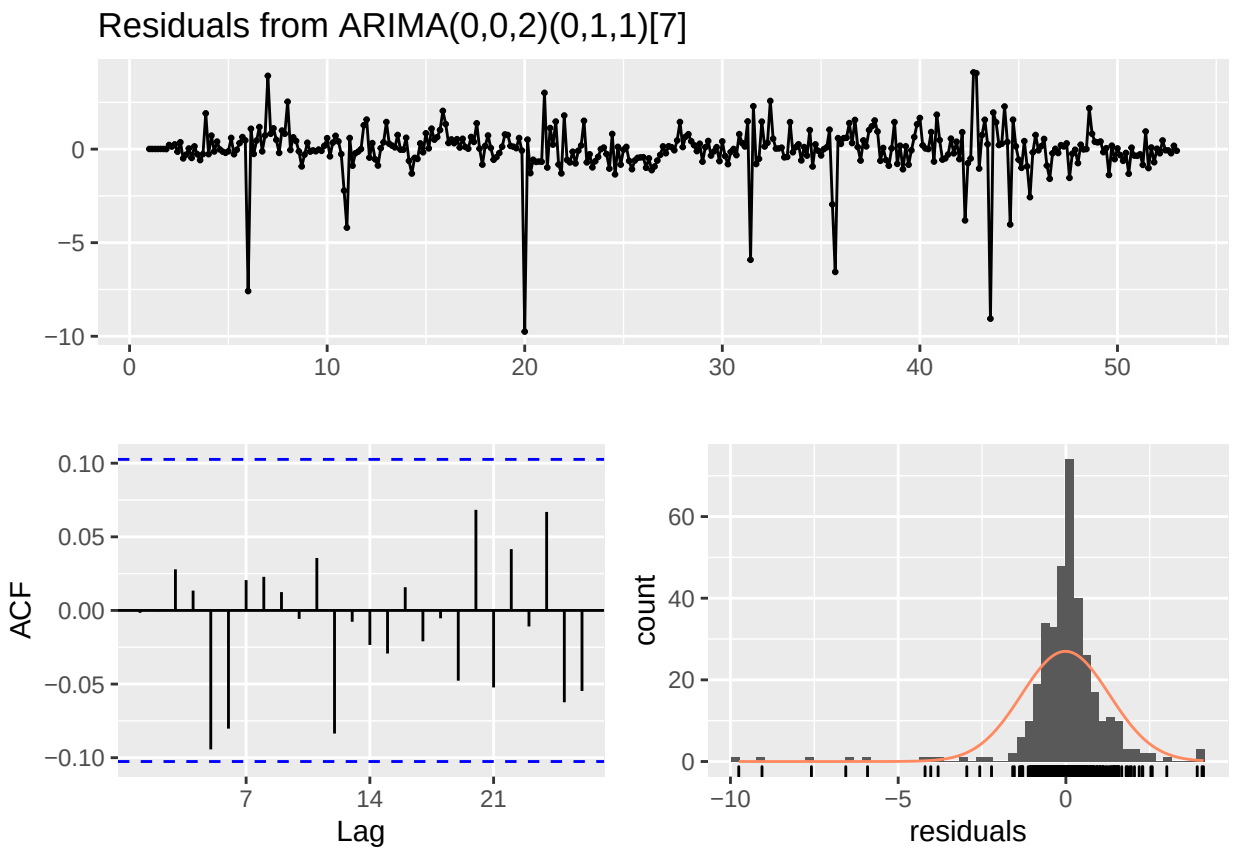
```
## ETS(A,N,A)
##
## Call:
##  ets(y = ., lambda = atm1.lambda)
##
##   Box-Cox transformation: lambda= 0.2616
##
##   Smoothing parameters:
##     alpha = 1e-04
##     gamma = 0.3513
##
##   Initial states:
##     l = 7.9717
##     s = -4.5094 0.5635 1.0854 0.5711 0.9551 0.5582
##           0.7761
##
##   sigma:  1.343
##
##      AIC     AICc      BIC
## 2379.653 2380.275 2418.652
```

```
atm1.ts %>% auto.arima(lambda = atm1.lambda )
```

```
## Series: .
## ARIMA(0,0,2)(0,1,1)[7]
## Box Cox transformation: lambda= 0.2615708
##
```

```
## Coefficients:
##          ma1      ma2     sma1
##       0.1126  -0.1094  -0.6418
## s.e.  0.0524   0.0520   0.0432
##
## sigma^2 estimated as 1.764:  log likelihood=-609.99
## AIC=1227.98   AICc=1228.09   BIC=1243.5
```

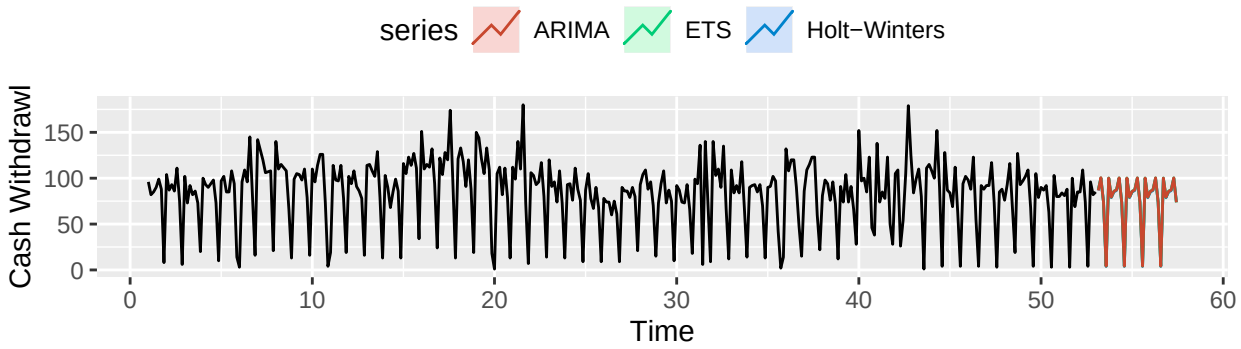
```
checkresiduals(atm1.ts %>% auto.arima(lambda = atm1.lambda ))
```



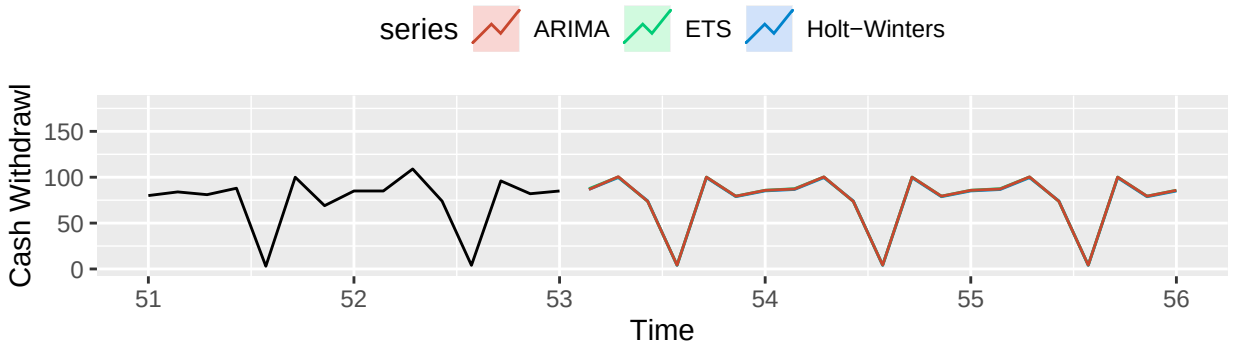

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,2)(0,1,1)[7]
## Q* = 9.8626, df = 11, p-value = 0.5428
##
## Model df: 3.   Total lags used: 14
```

```
atm.forecast(atm1.ts)
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```



Zoom in

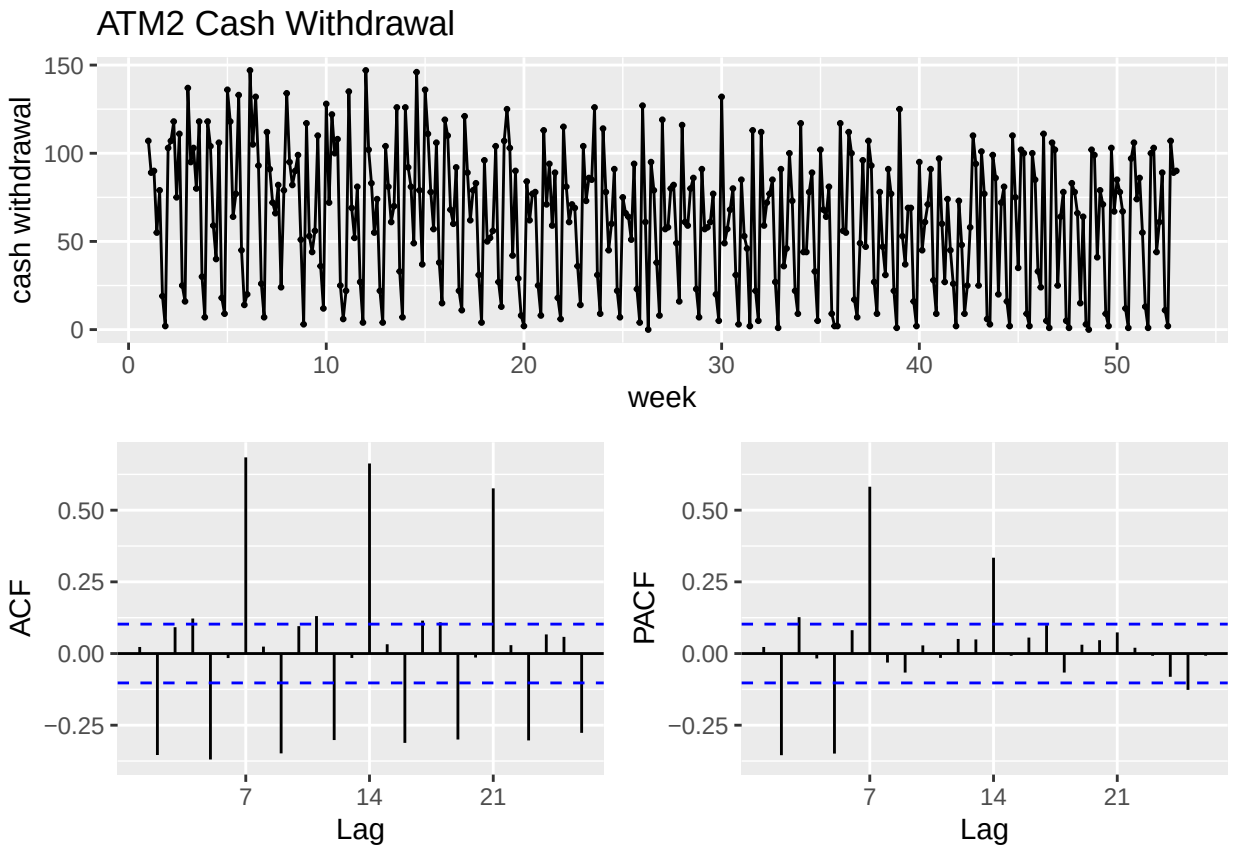



```
model_accuracy(atm1.ts)
```

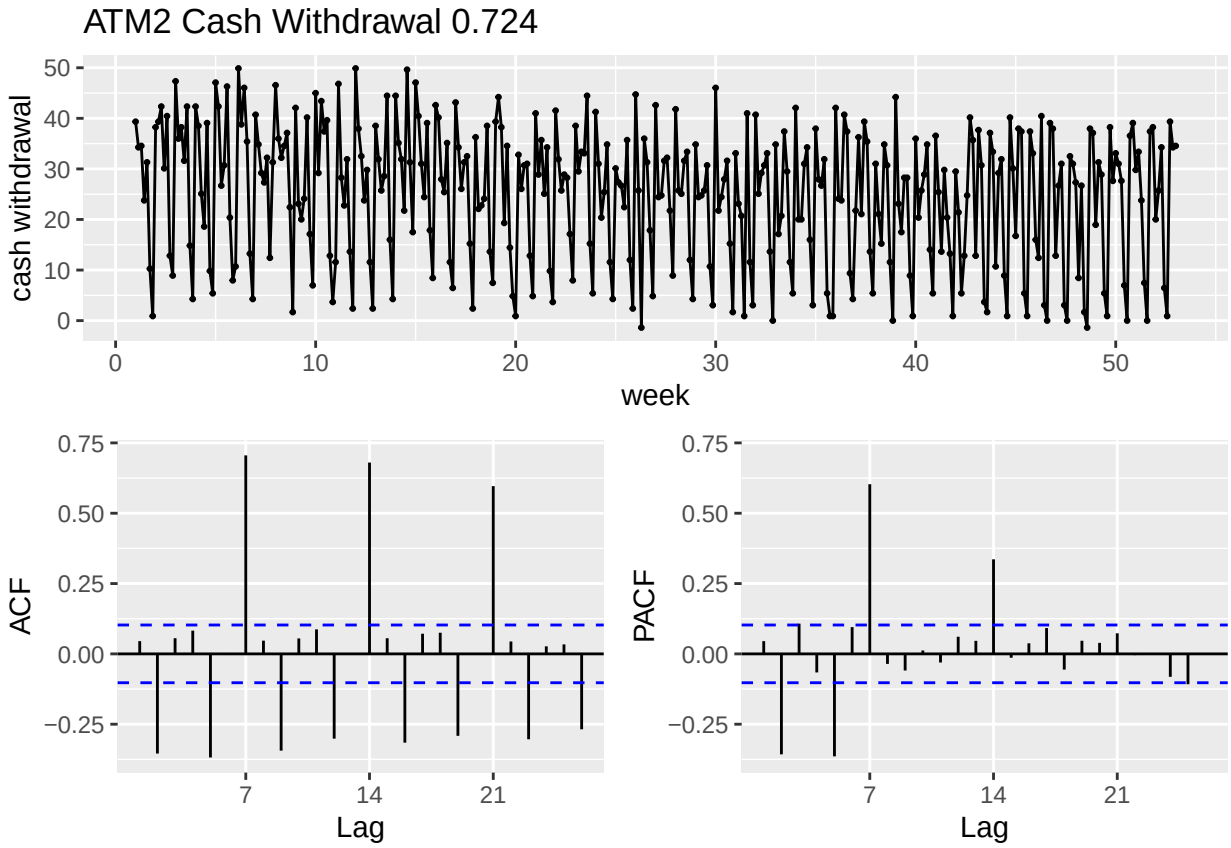
```
##   Holt-Winters      ETS    ARIMA
## 1     25.24631 24.92166 24.93069
```

ATM2

```
atm2.ts <- atm.new %>% filter(ATM=="ATM2") %>% select(Cash) %>% ts(frequency = 7)
ggtsdisplay(atm2.ts, main="ATM2 Cash Withdrawal", ylab="cash withdrawal", xlab="week")
```




```
atm2.lambda <- BoxCox.lambda(atm2.ts)
atm2.ts.bc <- BoxCox(atm2.ts, atm2.lambda )
ggtsdisplay(atm2.ts.bc, main=paste("ATM2 Cash Withdrawal",round(atm2.lambda, 3)), ylab="cash withdrawal
```




```
# Number of differences required for a stationary series
ndiffs(atm2.ts.bc)
```

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

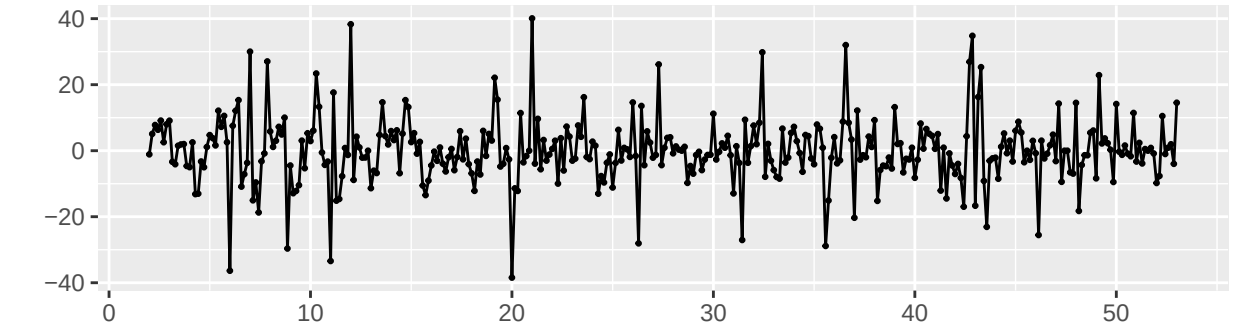
```
# Number of differences required for a seasonally stationary series
nsdiffs(atm2.ts.bc)
```

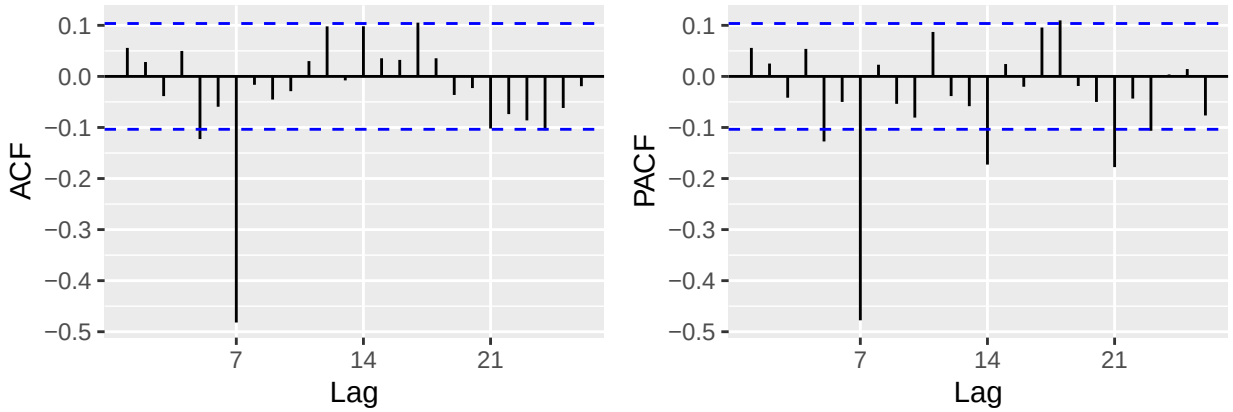
```
## [1] 1
```

```
atm2.ts.bc %>% diff(lag=7) %>% ur.kpss() %>% summary()
```

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.0162
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

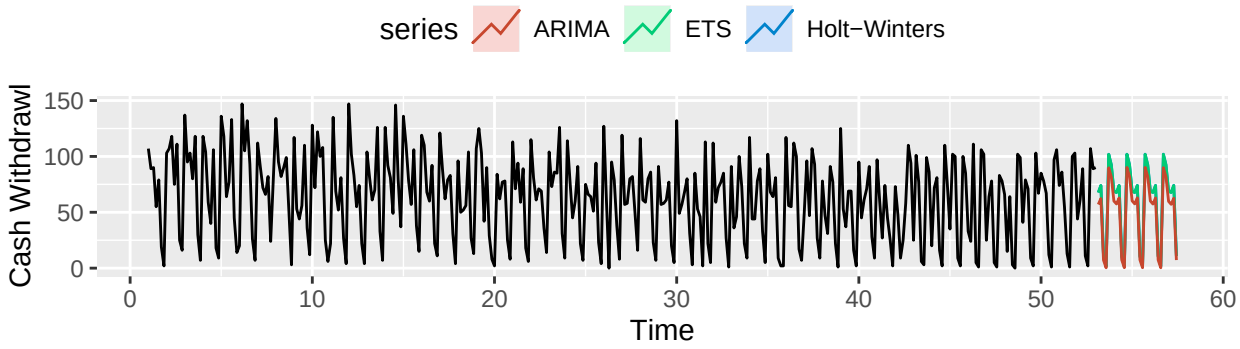
```
atm2.ts.bc %>% diff(lag=7) %>% ggtsdisplay()
```



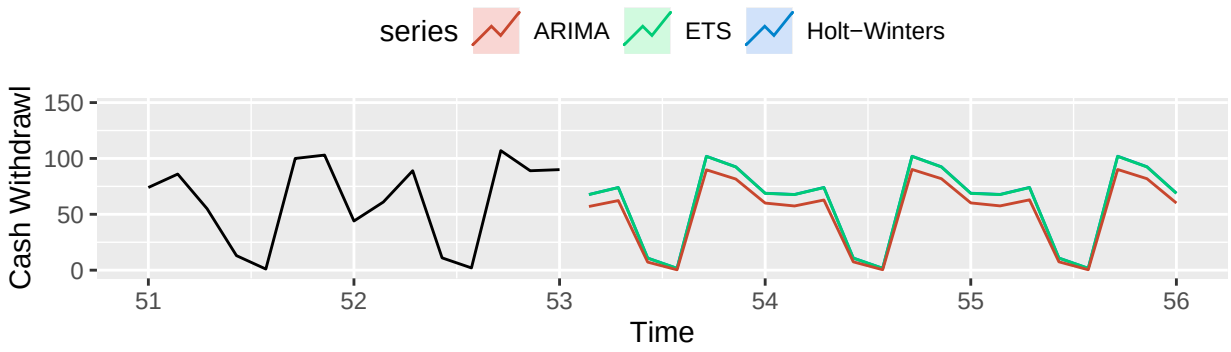



```
atm.forecast(atm2.ts)
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```



Zoom in

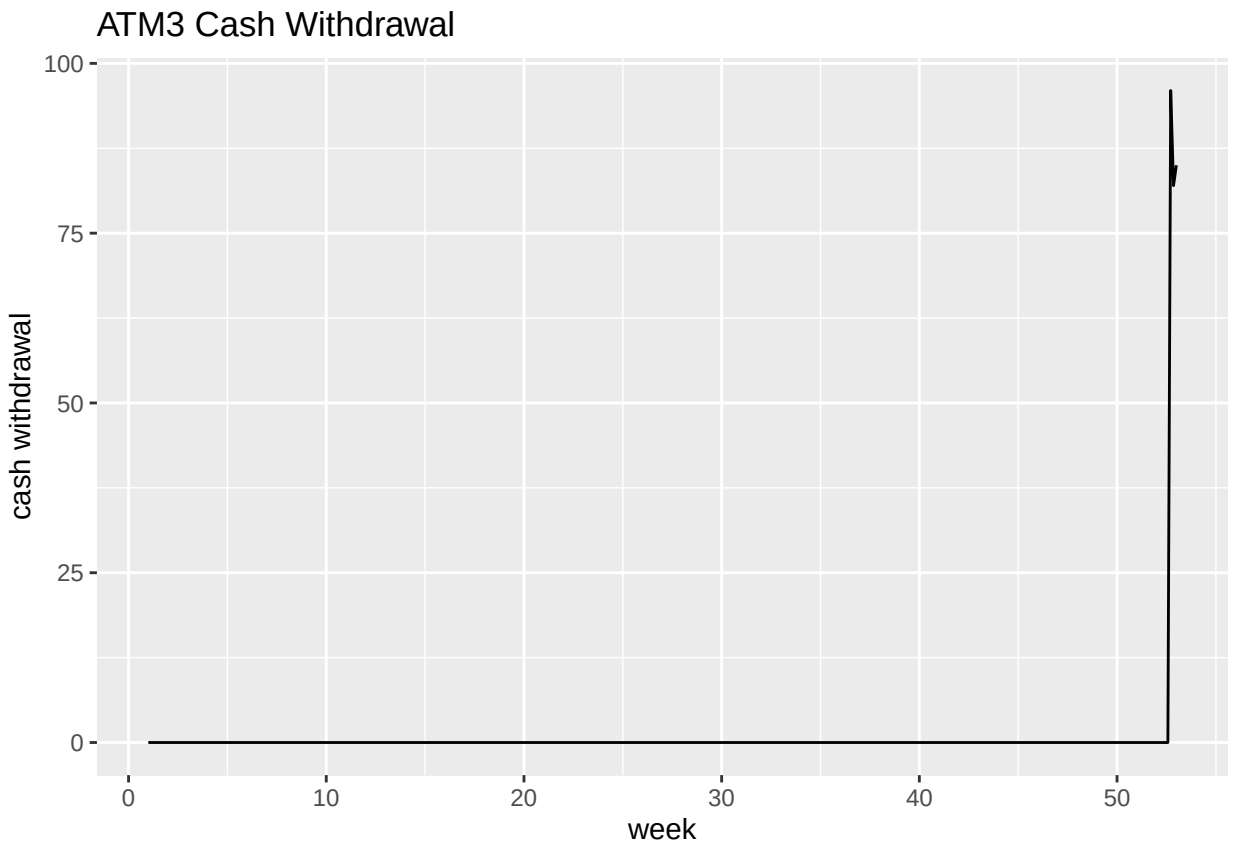


```
model_accuracy(atm2.ts)
```

```
##   Holt-Winters      ETS    ARIMA
## 1     25.44307 25.35721 24.27083
```

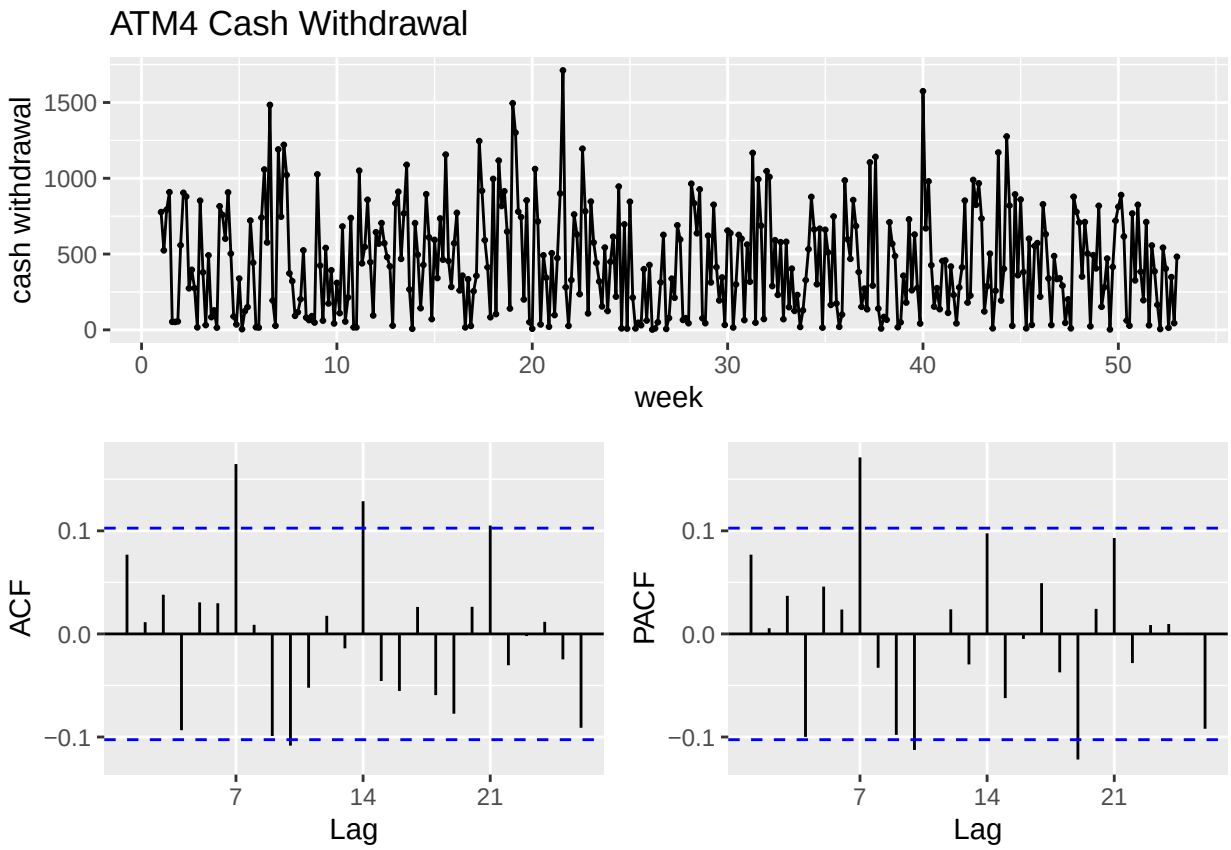
ATM3

```
atm3.ts <- atm.new %>% filter(ATM=="ATM3") %>% select(Cash) %>% ts(frequency = 7)
autoplot(atm3.ts, main="ATM3 Cash Withdrawal", ylab="cash withdrawal", xlab="week")
```

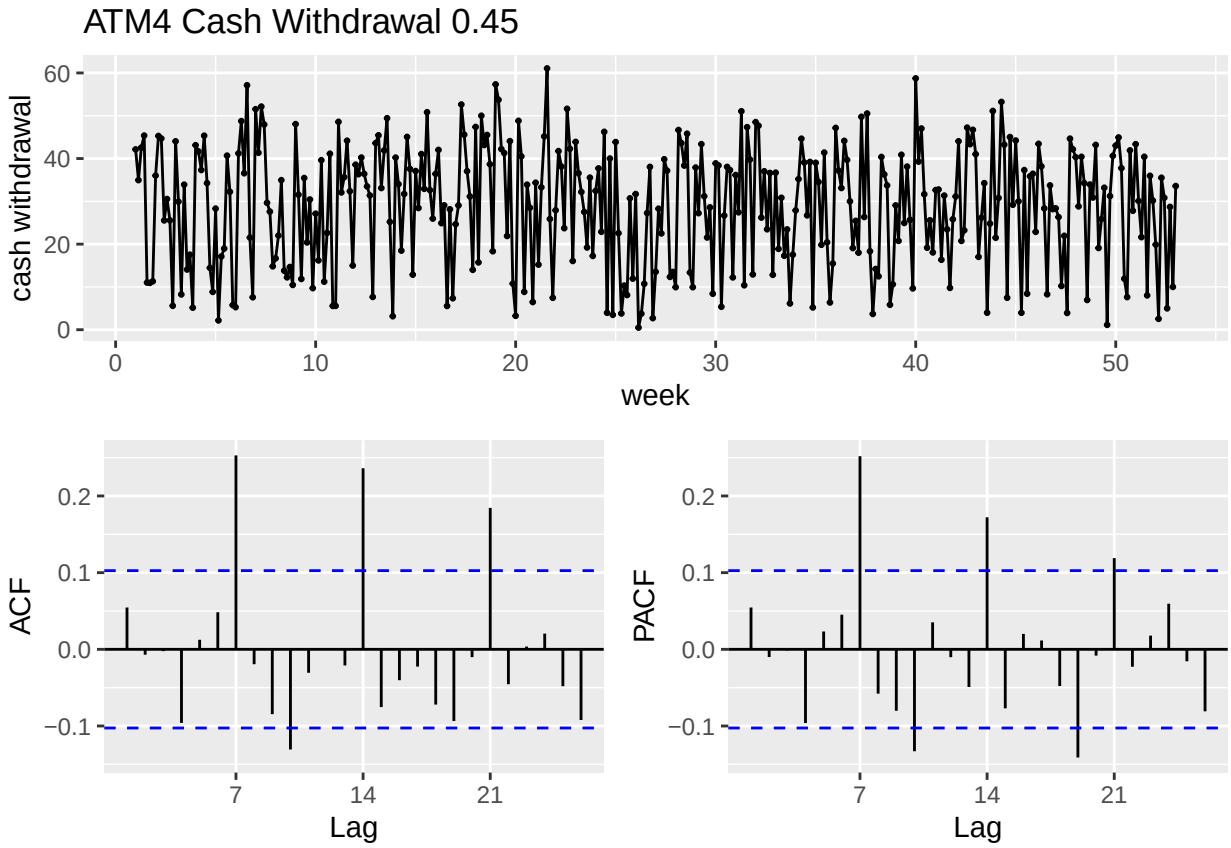


ATM4

```
atm4.ts <- atm.new %>% filter(ATM=="ATM4") %>% select(Cash) %>% ts(frequency = 7)
ggtsdisplay(atm4.ts, main="ATM4 Cash Withdrawal", ylab="cash withdrawal", xlab="week")
```




```
atm4.lambda <- BoxCox.lambda(atm4.ts)
atm4.ts.bc <- BoxCox(atm4.ts, atm4.lambda )
ggtsdisplay(atm4.ts.bc, main=paste("ATM4 Cash Withdrawal",round(atm4.lambda, 3)), ylab="cash withdrawal
```




```
# Number of differences required for a stationary series
ndiffs(atm4.ts.bc)
```

```
## [1] 0
```

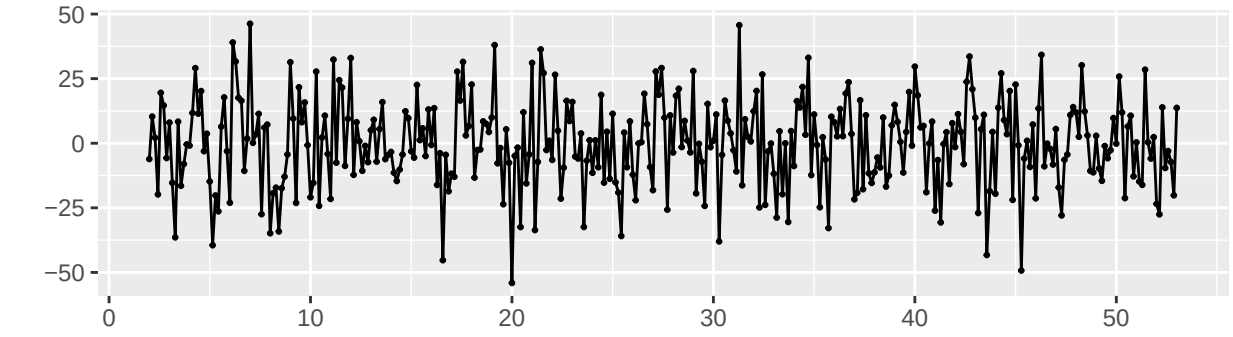
```
# Number of differences required for a seasonally stationary series
nsdiffs(atm4.ts.bc)
```

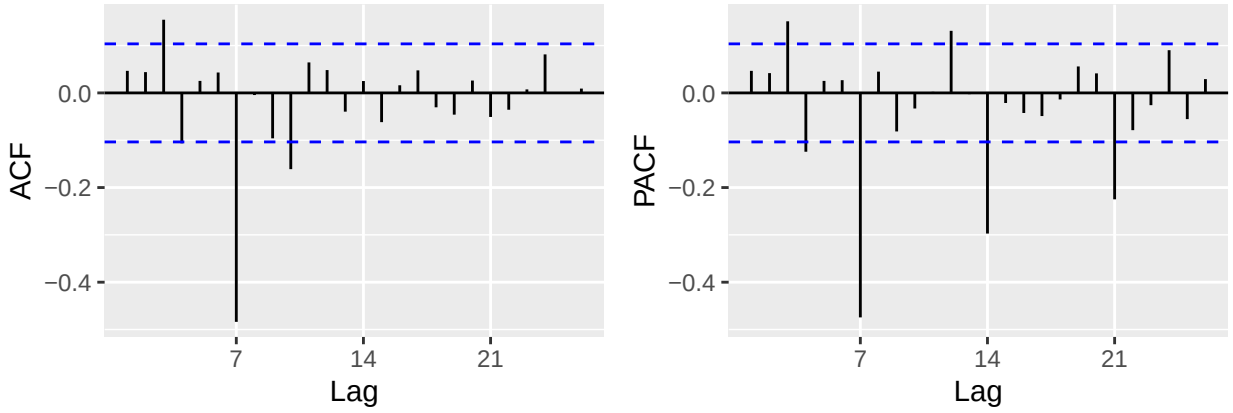
```
## [1] 0
```

```
atm4.ts.bc %>% ur.kpss() %>% summary()
```

```
##
## ########################
## # KPSS Unit Root Test #
## ########################
##
## Test is of type: mu with 5 lags.
##
## Value of test-statistic is: 0.0792
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

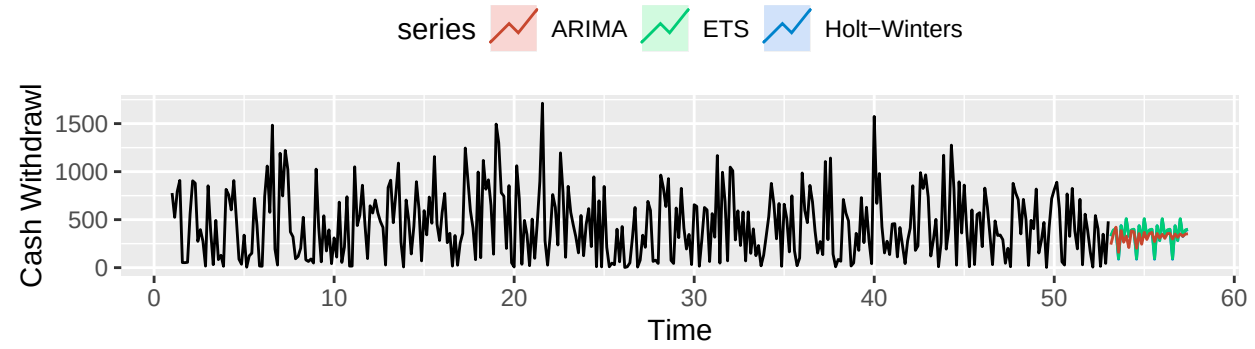
```
atm4.ts.bc %>% diff(lag=7) %>% ggtsdisplay()
```



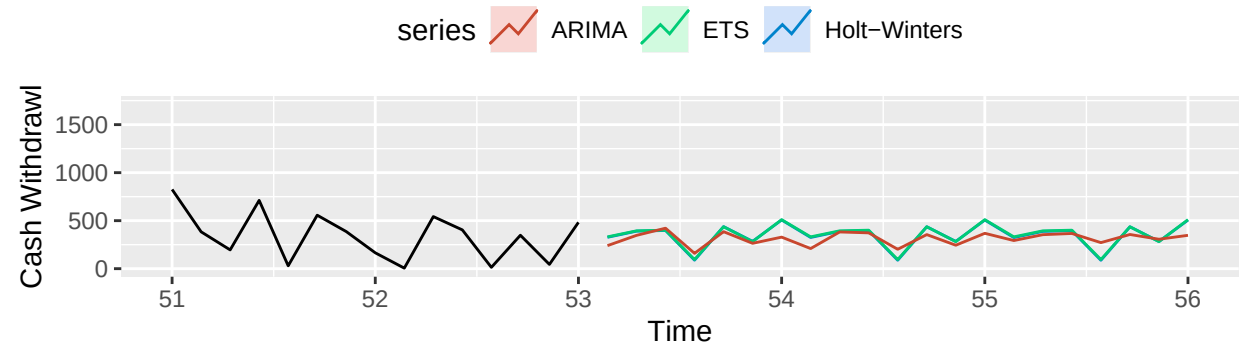



```
atm.forecast(atm4.ts)
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which will
## replace the existing scale.
```



Zoom in



```
model_accuracy(atm4.ts)
```

```
##   Holt-Winters      ETS    ARIMA
## 1     340.8111 337.9663 352.0876
```

```
atm4.ts %>% ets(lambda = BoxCox.lambda(atm4.ts))
```

```
## ETS(A,N,A)
##
## Call:
##  ets(y = ., lambda = BoxCox.lambda(atm4.ts))
##
##   Box-Cox transformation: lambda= 0.4498
##
##   Smoothing parameters:
##     alpha = 1e-04
##     gamma = 0.1035
##
##   Initial states:
##     l = 28.6369
##     s = -18.6503 -3.3529 1.6831 4.7437 5.4471 4.9022
##            5.2271
##
##   sigma:  12.9202
##
##      AIC     AICc      BIC
## 4032.268 4032.890 4071.267
```

```
atm4.ts %>% auto.arima(lambda = BoxCox.lambda(atm4.ts))
```

```
## Series: .
## ARIMA(0,0,1)(2,0,0)[7] with non-zero mean
## Box Cox transformation: lambda= 0.449771
##
## Coefficients:
##          ma1    sar1    sar2     mean
##       0.0790  0.2078  0.2023  28.6364
## s.e.  0.0527  0.0516  0.0525   1.2405
##
## sigma^2 estimated as 176.5:  log likelihood=-1460.57
## AIC=2931.14   AICc=2931.3   BIC=2950.64
```

Part B - Forecasting Power

```
download.file(
  url="https://github.com/amit-kapoor/data624/blob/main/Project1/ResidentialCustomerForecastLoad-624.xls
  destfile = temp.file,
  mode = "wb",
  quiet = TRUE)
power.data <- read_excel(temp.file, skip=0, col_types = c("numeric","text","numeric"))

head(power.data)
```

```
## # A tibble: 6 x 3
##   CaseSequence `YYYY-MMM`     KWH
```

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

Part C - Waterflow Pipe

```
download.file(url="https://github.com/amit-kapoor/data624/blob/main/Project1/Waterflow_Pipe1.xlsx?raw=t
              destfile = temp.file,
              mode = "wb",
              quiet = TRUE)
pipe1.data <- read_excel(temp.file, skip=0, col_types = c("date","numeric"))

download.file(url="https://github.com/amit-kapoor/data624/blob/main/Project1/Waterflow_Pipe2.xlsx?raw=t
              destfile = temp.file,
              mode = "wb",
              quiet = TRUE)

pipe2.data <- read_excel(temp.file, skip=0, col_types = c("date","numeric"))
head(pipe1.data)
```

```
## # A tibble: 6 x 2
##   `Date Time`         WaterFlow
##   <dttm>                  <dbl>
## 1 2015-10-23 00:24:06     23.4
## 2 2015-10-23 00:40:02     28.0
## 3 2015-10-23 00:53:51     23.1
## 4 2015-10-23 00:55:40     30.0
## 5 2015-10-23 01:19:17      6.00
## 6 2015-10-23 01:23:58     15.9
```