HA_D624_P1

Hantz Angrand

October 12, 2019

```
#Upload library
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0

## Warning: package 'ggplot2' was built under R version 3.5.2

## -- Conflicts ------------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
library(fpp2)
```

```
## Warning: package 'fpp2' was built under R version 3.5.3

## Loading required package: forecast

## Warning: package 'forecast' was built under R version 3.5.3

## Loading required package: fma

## Warning: package 'fma' was built under R version 3.5.3

## Loading required package: expsmooth

## Warning: package 'expsmooth' was built under R version 3.5.3
```

```
library(forecast)
```

```
atm<-read_excel("C:/Users/hangr/Documents/fall2019/Data624/ATM624Data.xlsx")
head(atm,5)
```

```
## # A tibble: 5 x 3
##   DATE                ATM    Cash
##   <dttm>              <chr> <dbl>
## 1 2009-05-01 00:00:00 ATM1     96
## 2 2009-05-01 00:00:00 ATM2    107
## 3 2009-05-02 00:00:00 ATM1     82
## 4 2009-05-02 00:00:00 ATM2     89
## 5 2009-05-03 00:00:00 ATM1     85
```

```
#Drop null values
atm<-atm %>%
  drop_na()
```

```
#Convert each ATM to Column
atm<- atm %>%
```

```
  spread(ATM,Cash)
head(atm,5)
```

```
## # A tibble: 5 x 5
##   DATE                 ATM1  ATM2  ATM3  ATM4
##   <dttm>              <dbl> <dbl> <dbl> <dbl>
## 1 2009-05-01 00:00:00    96   107     0    96
## 2 2009-05-02 00:00:00    82    89     0    82
## 3 2009-05-03 00:00:00    85    90     0    85
## 4 2009-05-04 00:00:00    90    55     0    90
## 5 2009-05-05 00:00:00    99    79     0    99
```

```
#Fix the date column
atm <- atm %>%
  mutate(DATE =as.Date(DATE))
head(atm)
```

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

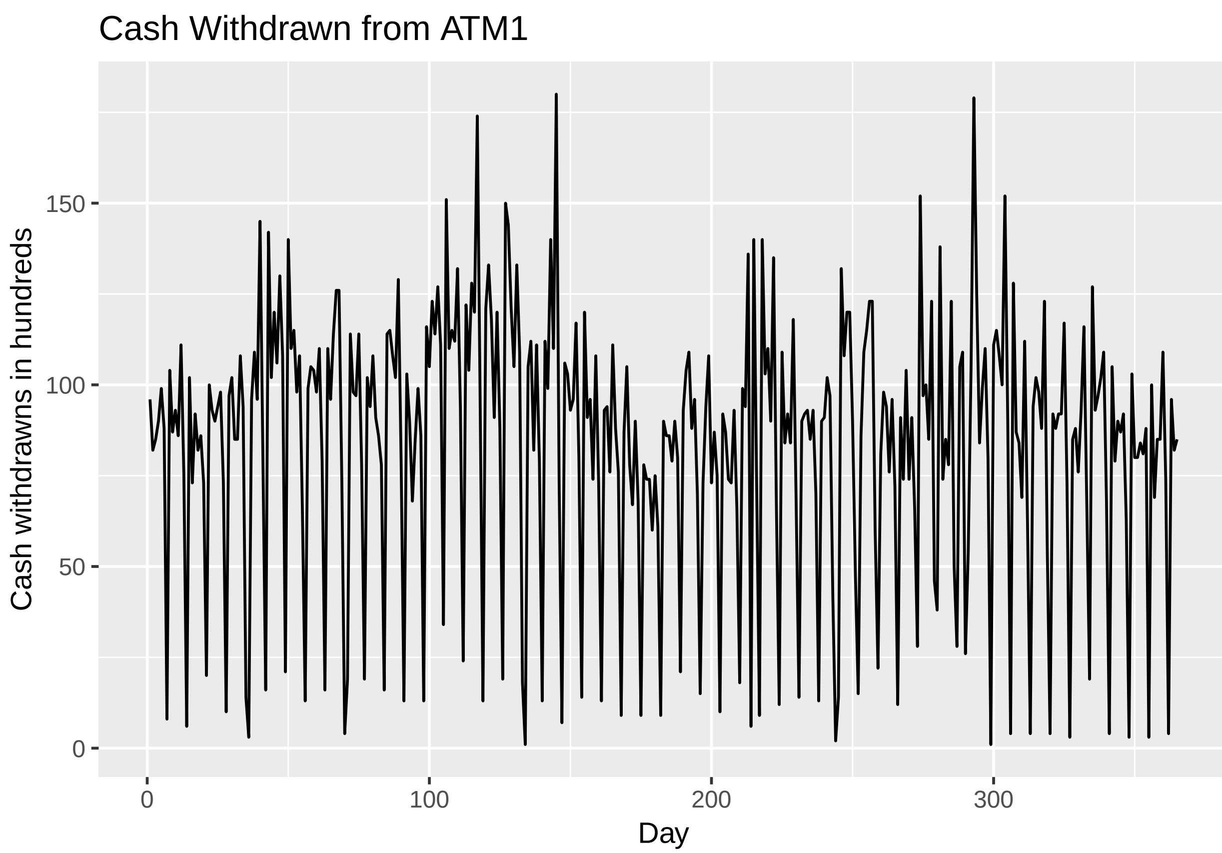
```
#Convert to a time series
ts_atm <- ts(atm %>% select(-DATE))

head(ts_atm)
```

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

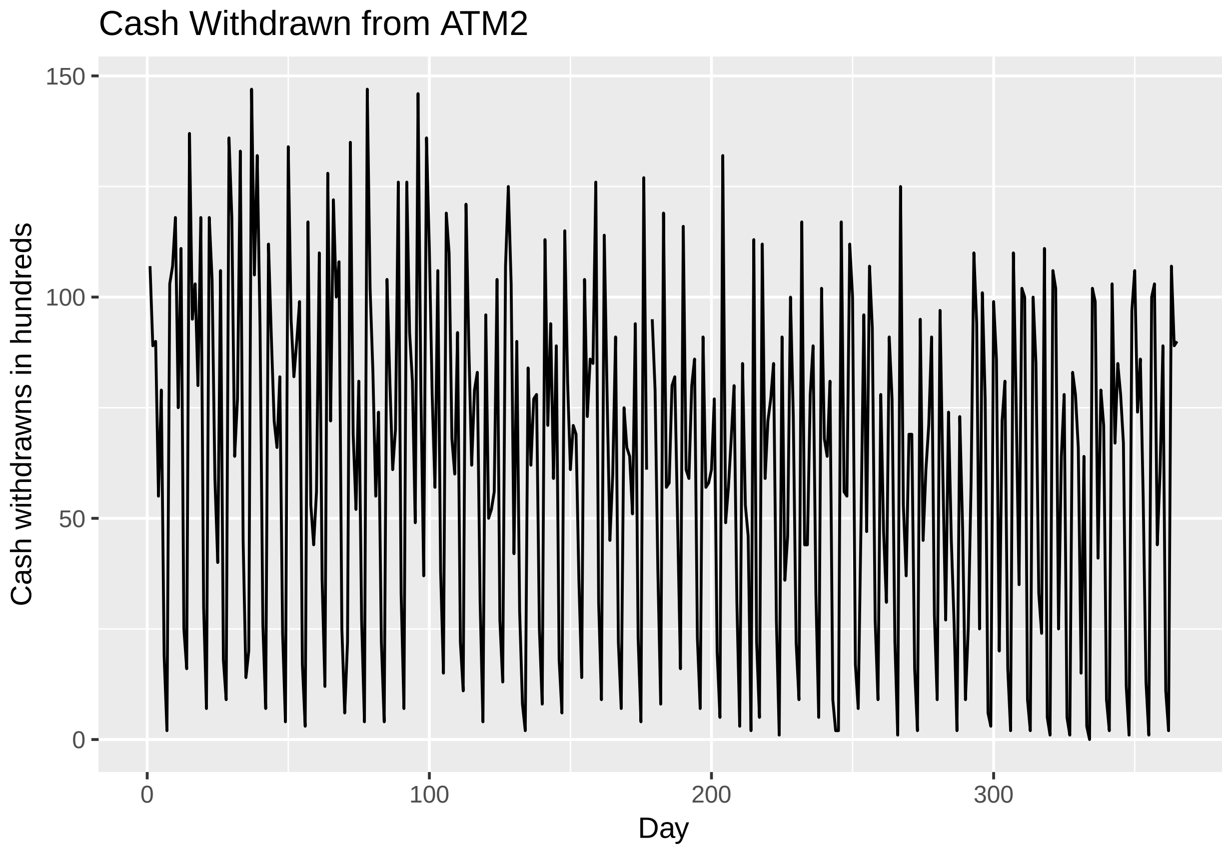
ATM1, ATM2, and ATM4 are a big deal of variation. buT ATM3 shows no cash withdrawn for most of the year. One assumption we can do about ATM3 is that it has just opened.

we will use the entire time series of ATM1 and ATM2. ATM3 will be used to forecast future prediction.

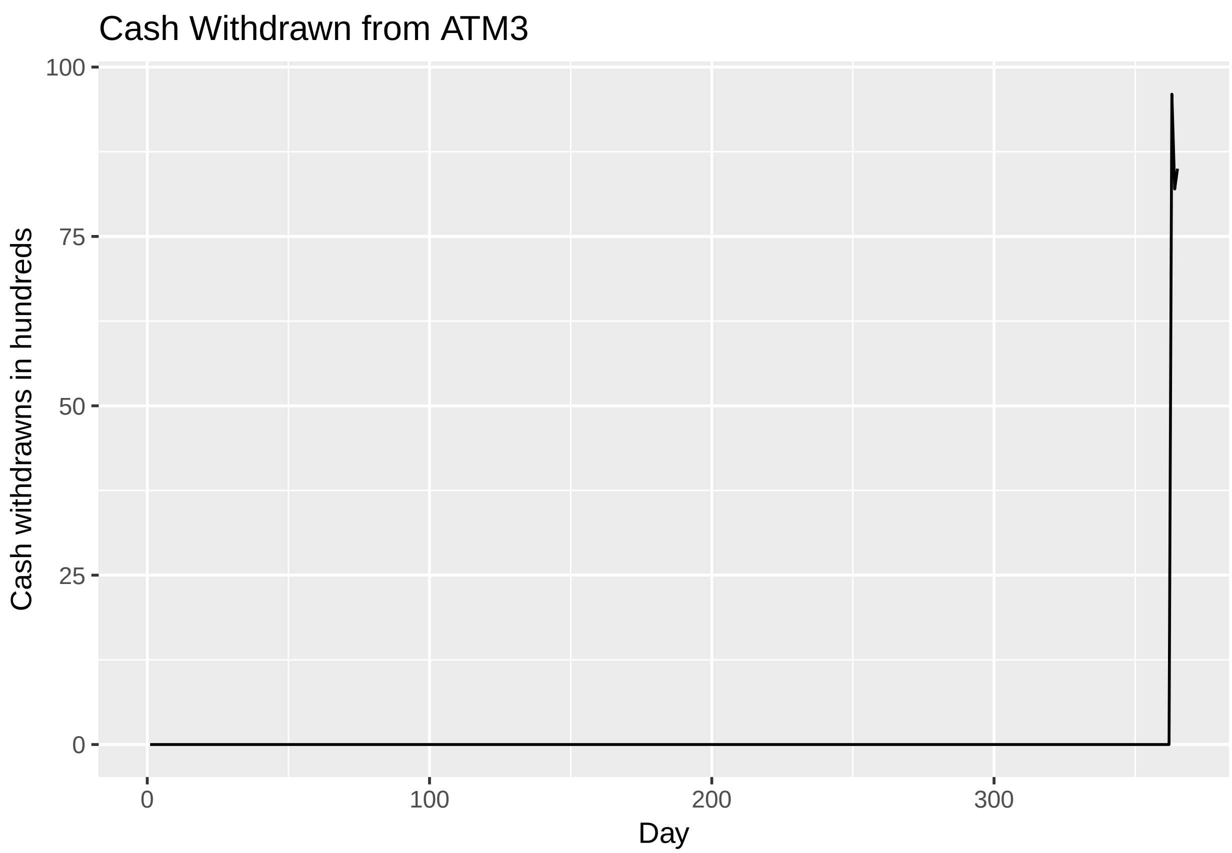
```
#Separate each ATM from the dataset and graph each dataset
atm1<-ts_atm[,"ATM1"]
autoplot(atm1) +
  labs(title ="Cash Withdrawn from ATM1", x="Day") +
  scale_y_continuous("Cash withdrawns in hundreds") +
  scale_color_discrete(NULL)
```



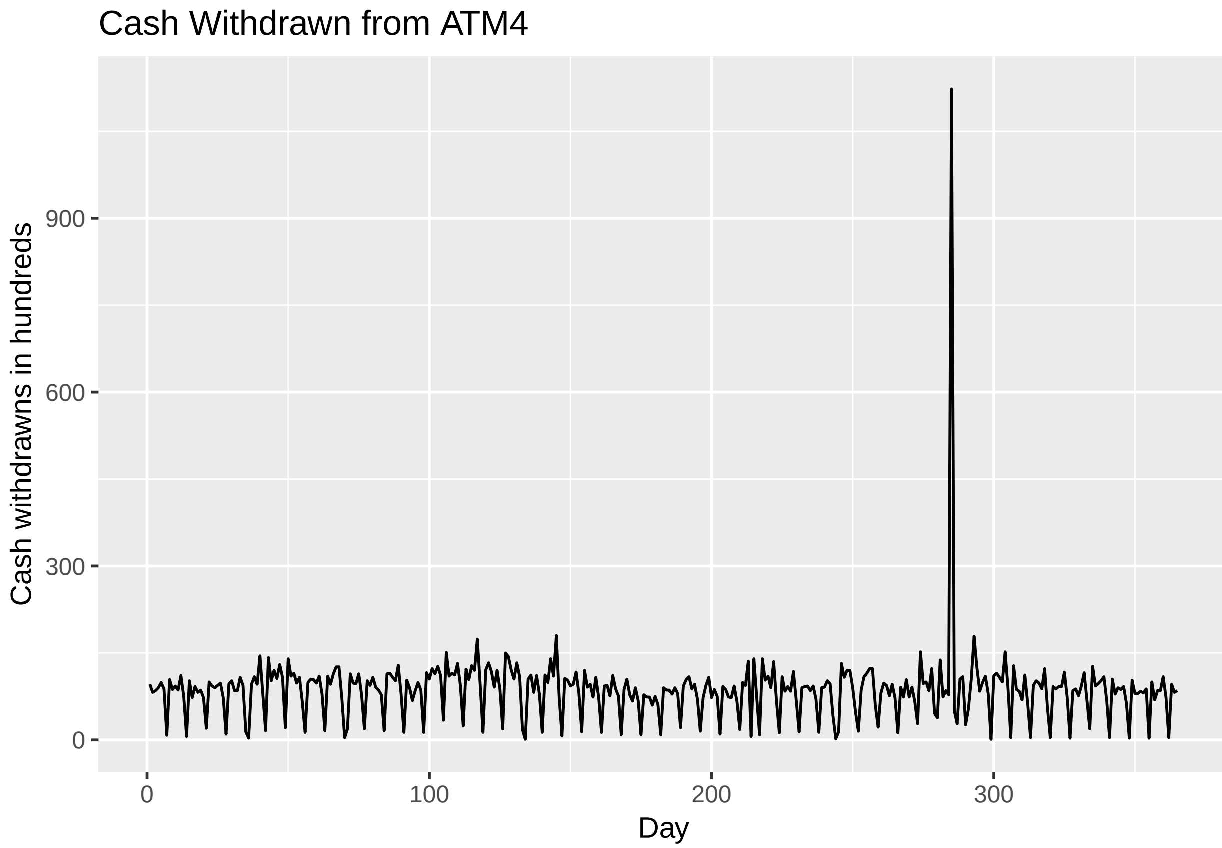

```r
#Separate each ATM from the dataset and graph each dataset
atm2<-ts_atm[,"ATM2"]
autoplot(atm2) +
  labs(title ="Cash Withdrawn from ATM2", x="Day") +
  scale_y_continuous("Cash withdrawns in hundreds") +
  scale_color_discrete(NULL)
```




```r
#Separate each ATM from the dataset and graph each dataset
atm3<-ts_atm[,"ATM3"]
autoplot(atm3) +
  labs(title ="Cash Withdrawn from ATM3", x="Day") +
  scale_y_continuous("Cash withdrawns in hundreds") +
  scale_color_discrete(NULL)
```

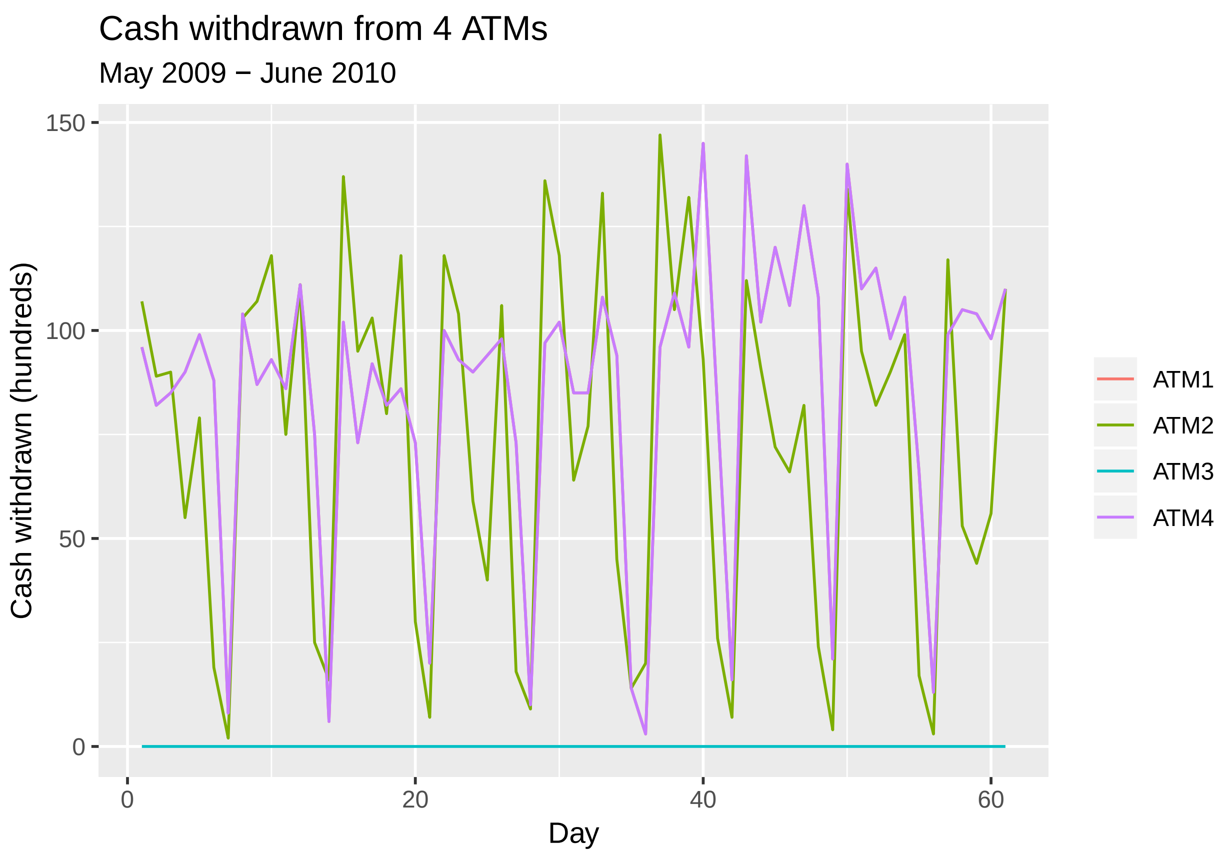



```
#Separate each ATM from the dataset and graph each dataset
atm4<-ts_atm[,"ATM4"]
autoplot(atm4) +
  labs(title ="Cash Withdrawn from ATM4", x="Day") +
  scale_y_continuous("Cash withdrawns in hundreds") +
  scale_color_discrete(NULL)
```



ATM1, ATM2 and ATM4 show a lot of deal of seasonality in the withdrawn from those ATM.We can further analyze it by selecting the first 2 months of the data.

```
autoplot(ts(ts_atm[1:61, ])) +

  labs(title = "Cash withdrawn from 4 ATMs",

       subtitle = "May 2009 - June 2010",

       x = "Day") +

  scale_y_continuous("Cash withdrawn (hundreds)") +

  scale_color_discrete(NULL)
```

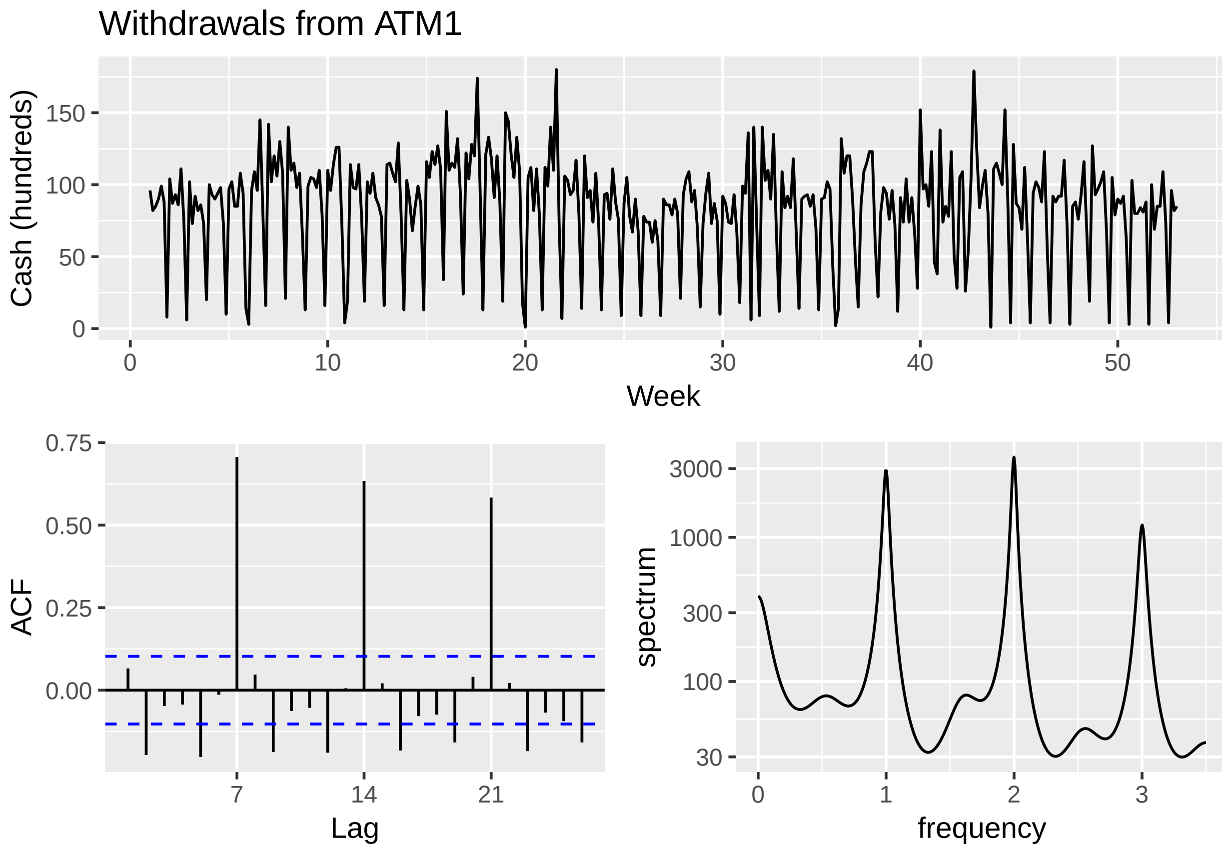


The data presents a sort of weekly seasonnality. To capture the seasonnality of this data we will set the frequency to 7.

```
atm1_freq<-ts(atm1, frequency =7)
atm2_freq<-ts(atm2, frequency=7)
atm4_freq<-ts(atm4, frequency=7)
```

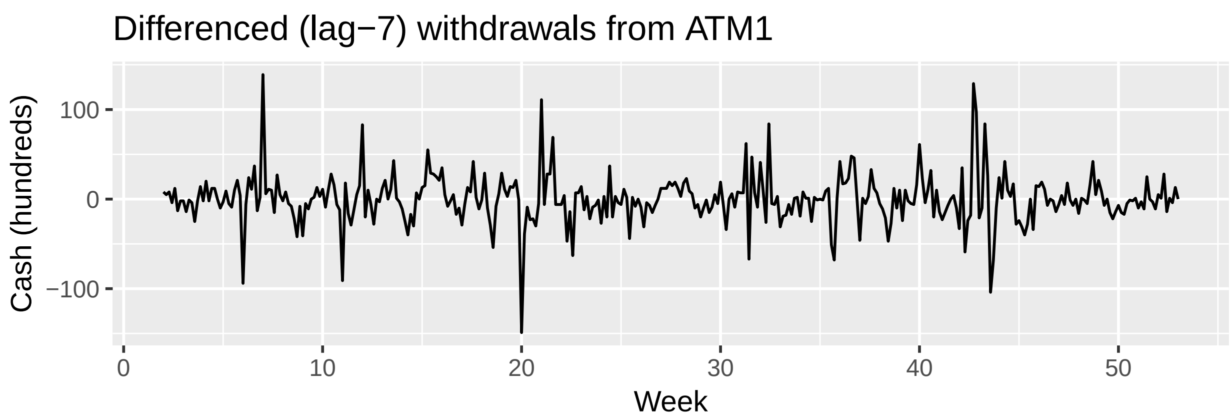
ATM1

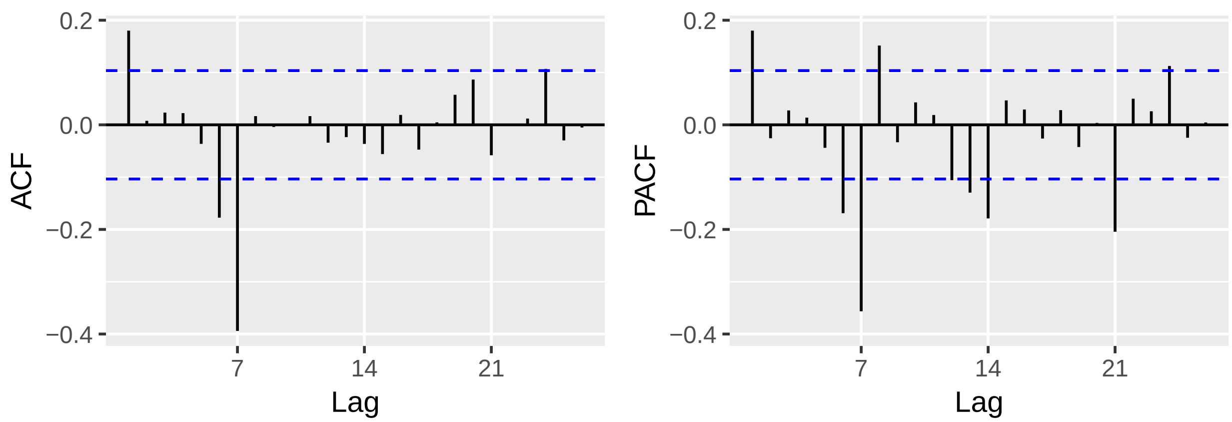
```
#ACF and spectrum plot
ggtsdisplay(atm1_freq, points = FALSE, plot.type = "spectrum",

            main = "Withdrawals from ATM1", xlab = "Week", ylab = "Cash (hundreds)")
```



In 7, 14 and 21 there are large spikes. the frequency 1,2,3 show the spike as well. Both suggest a seasonnal ARISMA model.

```
ggtsdisplay(diff(atm1_freq, 7), points = FALSE,

            main = "Differenced (lag-7) withdrawals from ATM1",

            xlab = "Week", ylab = "Cash (hundreds)")
```





BoxCox transformation to estimate lambda

```
# get optimal lambda for Box-cox transformation

lambda_atm1<- BoxCox.lambda(atm1_freq)

# define function to create models & return AIC values for timeseries

aic_atm<- function(p, d, q, P, D, Q) {

  # create model with Box-Cox and specified ARIMA parameters; extract AIC

  AIC(Arima(atm1_freq, order = c(p, d, q), seasonal = c(P, D, Q), lambda = lambda_atm1))

}

# create possible combinations of p, q, P, Q except all zero

expand.grid(p = 0:1, q = 0:1, P = 0:1, Q = 0:1) %>%

  filter(p > 0 | q > 0 | P > 0 | Q > 0) %>%

  # calc AIC for models

  mutate(aic = pmap_dbl(list(p, 0, q, P, 1, Q), aic_atm)) %>%

  # return best AIC
```

```
  slice(which.min(aic))
```

```
##   p q P Q     aic
## 1 1 1 0 1 1221.26
```

The minimum aic value is for non-seasonality AR(1) and MA(1). AR(0) and AM(1) is for seasonality. Let's fit the model using arima model arima(1,0,1)(0,1,1)

```
fit_atm1 <- Arima(atm1_freq, order = c(1, 0, 1), seasonal = c(0, 1, 1), lambda = lambda_atm1)
summary(fit_atm1)
```

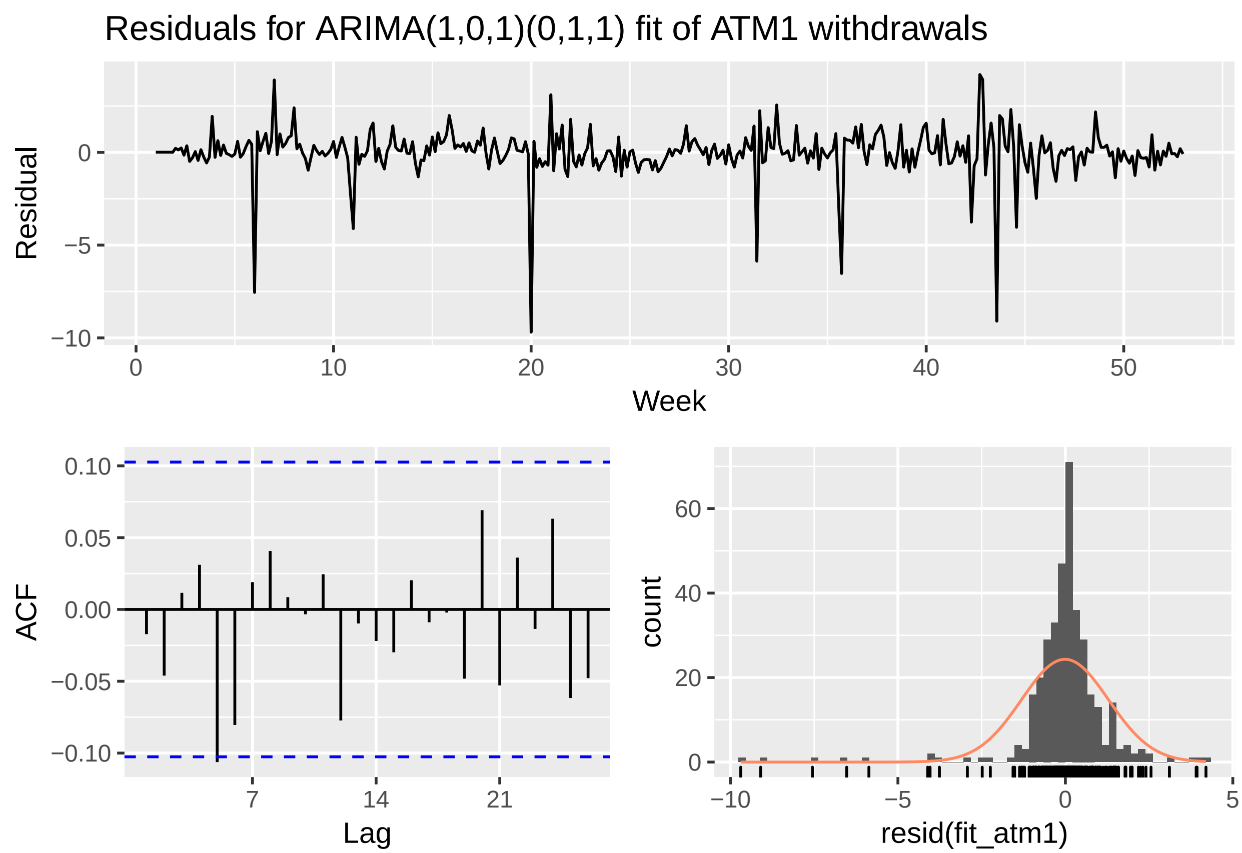
```
## Series: atm1_freq
## ARIMA(1,0,1)(0,1,1)[7]
## Box Cox transformation: lambda= 0.2584338
##
## Coefficients:
##           ar1     ma1     sma1
##       -0.4894  0.6125  -0.6385
## s.e.   0.2309  0.2081   0.0432
##
## sigma^2 estimated as 1.732:  log likelihood=-606.63
## AIC=1221.26   AICc=1221.37   BIC=1236.78
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE     MASE
## Training set 2.293003 24.81988 15.66437 -89.57546 108.1682 0.892827
##                      ACF1
## Training set -0.008839946
```

Let's diagnostic the residuals with Ljung-Box.

```
Box.test(resid(fit_atm1), type = "L", fitdf = 3, lag = 7)
```

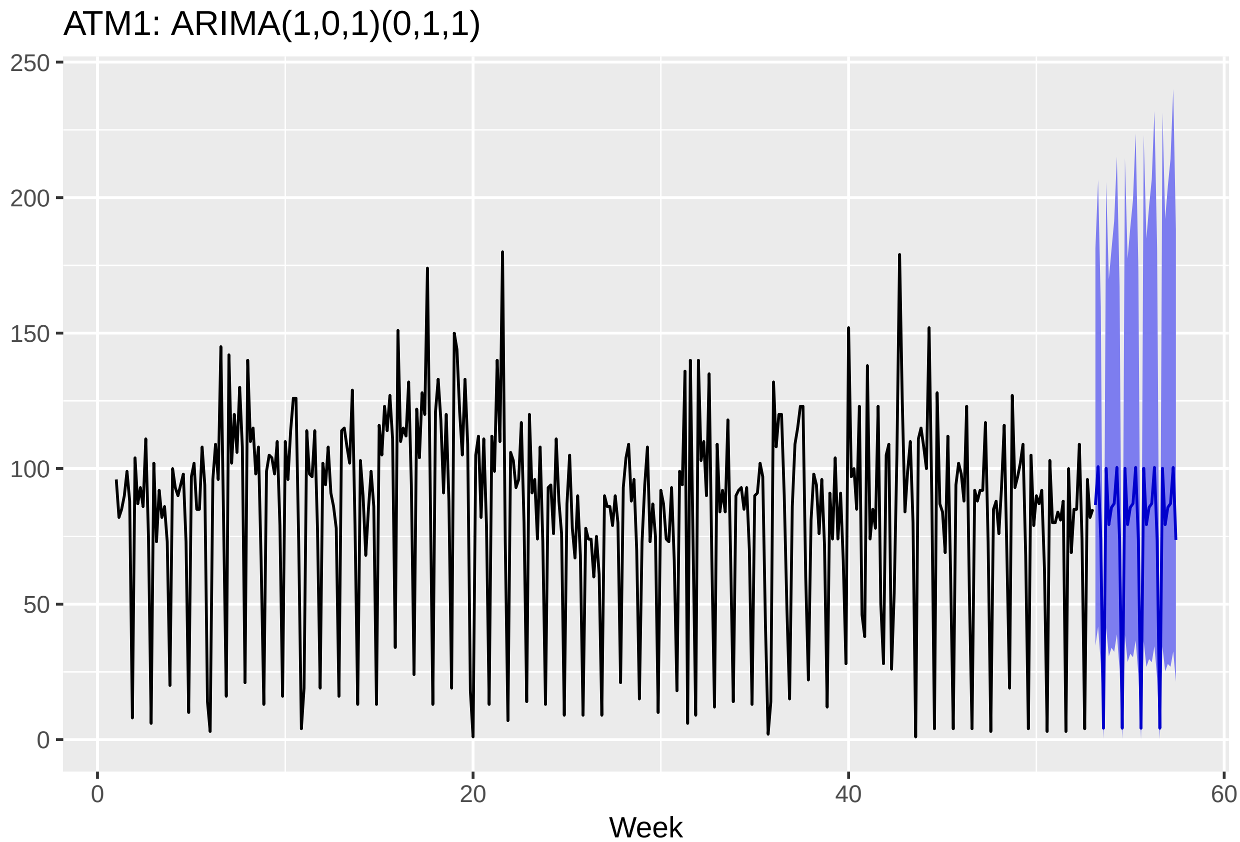
```
##
##  Box-Ljung test
##
## data:  resid(fit_atm1)
## X-squared = 8.0497, df = 4, p-value = 0.08977
```

```
ggtsdisplay(resid(fit_atm1), points = FALSE, plot.type = "histogram",

            main = "Residuals for ARIMA(1,0,1)(0,1,1) fit of ATM1 withdrawals",

            xlab = "Week", ylab = "Residual")
```



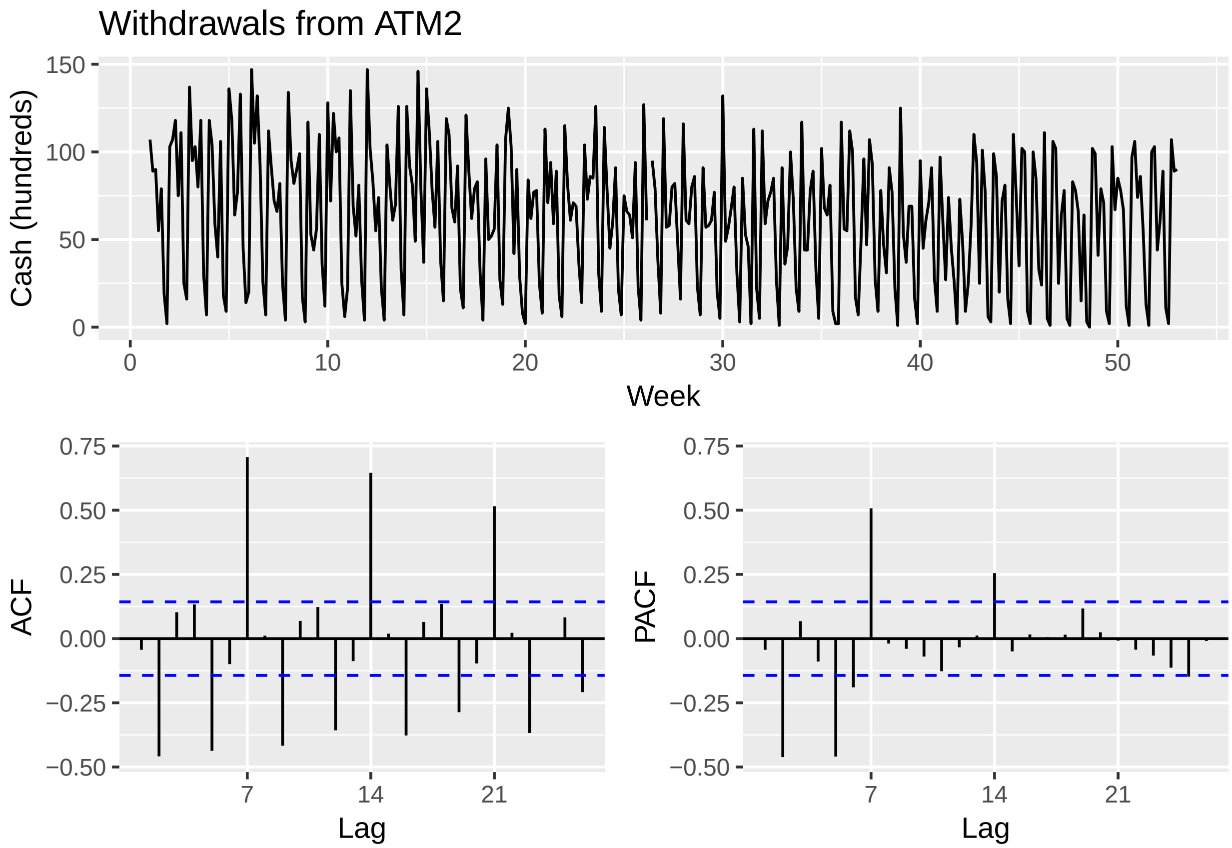
The p_value is greater than 0.05 meaning that the residual is white noise. The residuals are not correlated and there is a normal distribution around the mean 0. We can use that model for forecasting.

```
forecast_atm1 <- forecast(fit_atm1, 31, level = 95)
autoplot(forecast_atm1) +

    labs(title = "ATM1: ARIMA(1,0,1)(0,1,1)", x = "Week", y = NULL) +

    theme(legend.position = "none")
```



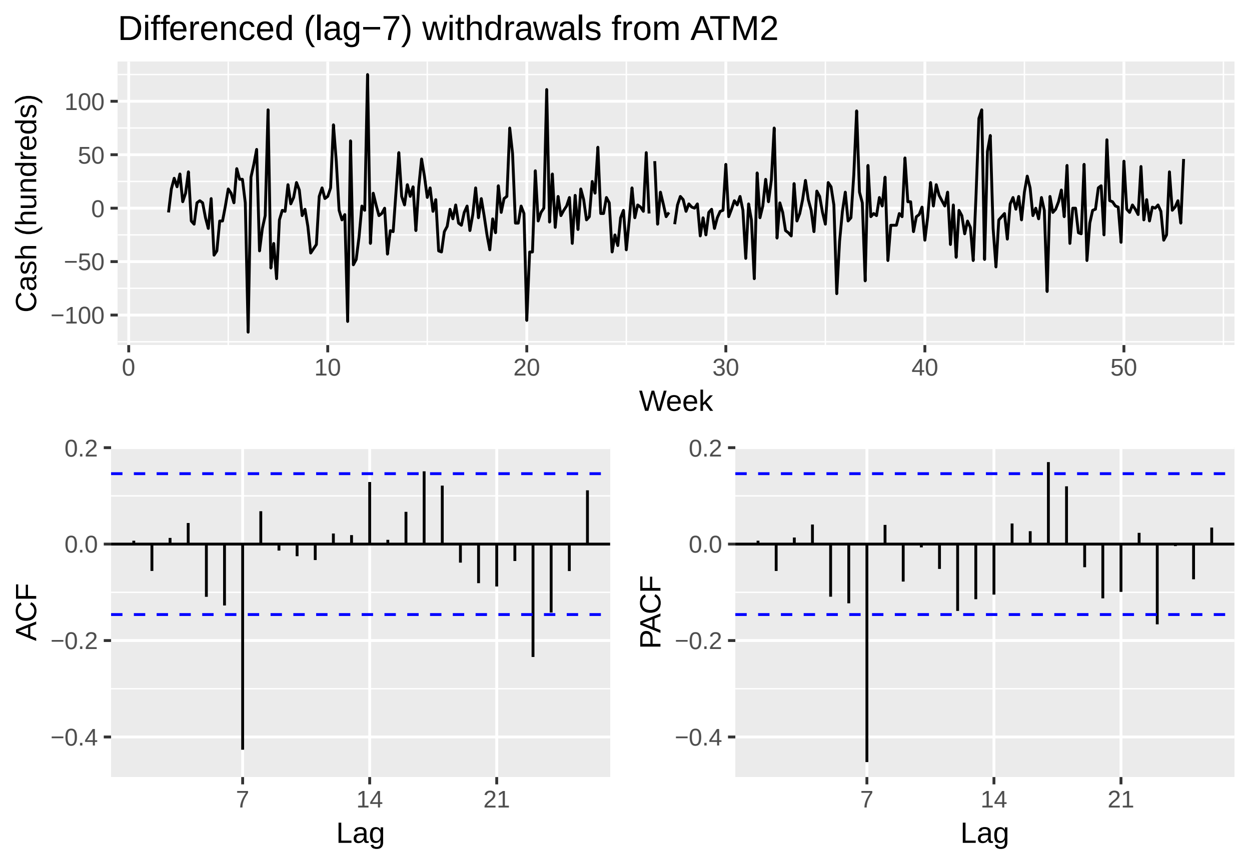
##ATM2 We can repeat the same stepp for ATM2.

```
ggtsdisplay(atm2_freq, points = FALSE,

            main = "Withdrawals from ATM2", xlab = "Week", ylab = "Cash (hundreds)")
```



The lag difference is 7.

```
ggtsdisplay(diff(atm2_freq, 7), points = FALSE,

            main = "Differenced (lag-7) withdrawals from ATM2",

            xlab = "Week", ylab = "Cash (hundreds)")
```



The spikes in ACF & PACF in the non-differenced series at $k = 2$ & $k = 5$ suggest $p, q \in [0, 2, 5]$. using the same aic function we can evaluate the minimum aic

```
# get optimal lambda for Box-cox transformation
lambda_atm2 <- BoxCox.lambda(atm2_freq)
# Evaluate aic
aic_atm <- function(p, d, q, P, D, Q) {
  # create model with Box-Cox and specified ARIMA parameters; extract AIC
  AIC(Arima(atm2_freq, order = c(p, d, q), seasonal = c(P, D, Q), lambda = lambda_atm2))
}
# create possible combinations of p, q, P, Q except all zero
expand.grid(p = c(0, 2, 5), q = c(0, 2, 5), P = 0:1, Q = 0:1) %>%
  filter(p > 0 | q > 0 | P > 0 | Q > 0) %>%
  # calculate AIC for models
  mutate(aic = pmap_dbl(list(p, 0, q, P, 1, Q), aic_atm)) %>%
```

```r
  # return minimum AIC

  slice(which.min(aic))
```

```
##   p q P Q      aic
## 1 2 2 0 1 2323.517
```

the model arima used is arima(5,0,5)(0,1,1). Let's fit that model

```r
fit_atm2<-Arima(atm2_freq, order = c(5, 0, 5), seasonal = c(0, 1, 1), lambda = lambda_atm2)
summary(fit_atm2)
```

```
## Series: atm2_freq
## ARIMA(5,0,5)(0,1,1)[7]
## Box Cox transformation: lambda= 0.6584081
##
## Coefficients:
##          ar1      ar2     ar3     ar4      ar5      ma1     ma2      ma3
##       0.2055  -0.1209  0.2260  0.3032  -0.4312  -0.1448  0.0114  -0.2213
## s.e.  0.4529   0.4033  0.2176  0.2419   0.4136   0.4787  0.4200   0.2100
##           ma4     ma5     sma1
##       -0.2466  0.2470  -0.6905
## s.e.   0.2463  0.4176   0.0595
##
## sigma^2 estimated as 37.91:  log likelihood=-1152.1
## AIC=2328.19   AICc=2329.1   BIC=2374.76
##
## Training set error measures:
##                     ME     RMSE      MAE  MPE MAPE      MASE        ACF1
## Training set 0.2238867 23.87153 16.65107 -Inf  Inf 0.8279025 -0.03050682
```

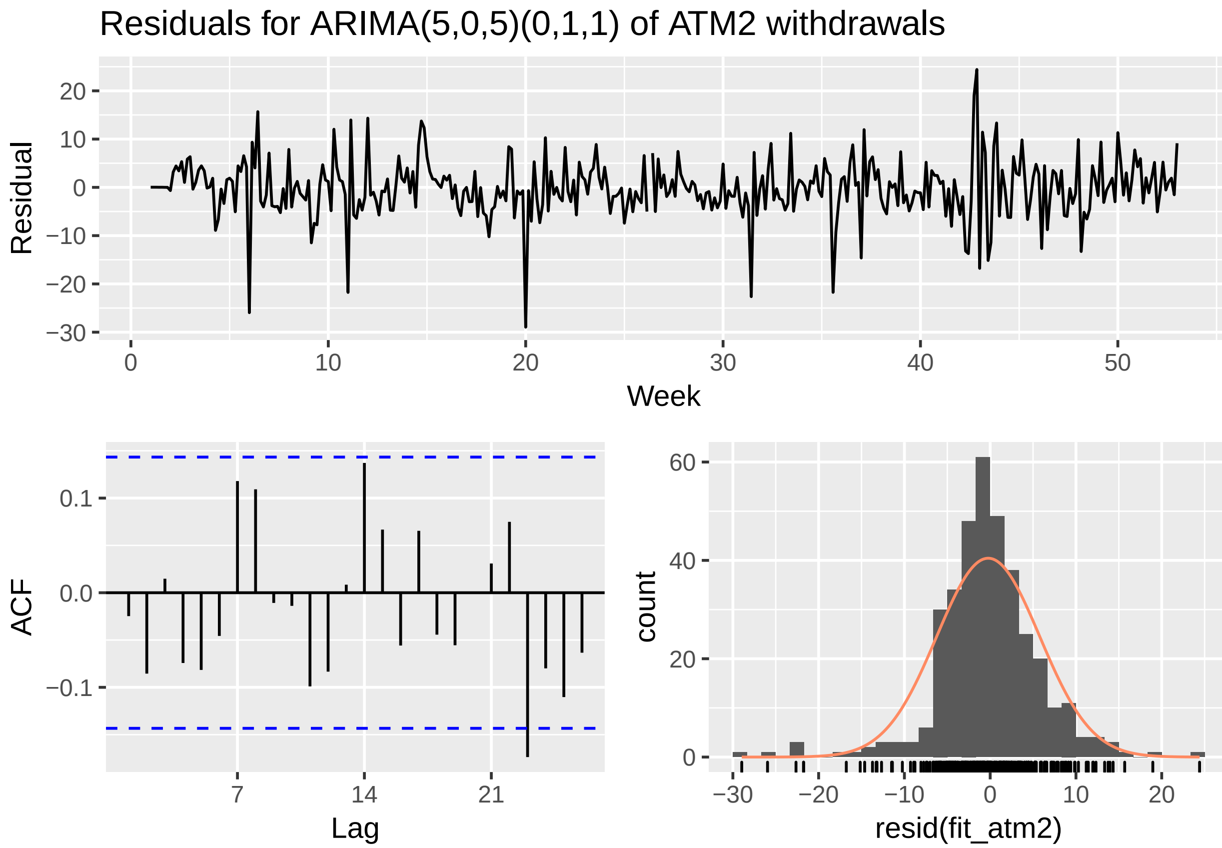
Let's evaluate the residual to check the validity of the model

```r
Box.test(resid(fit_atm2), type = "L", fitdf = 11, lag = 14)
```

```
##
##  Box-Ljung test
##
## data:  resid(fit_atm2)
## X-squared = 2.1119, df = 3, p-value = 0.5495
```

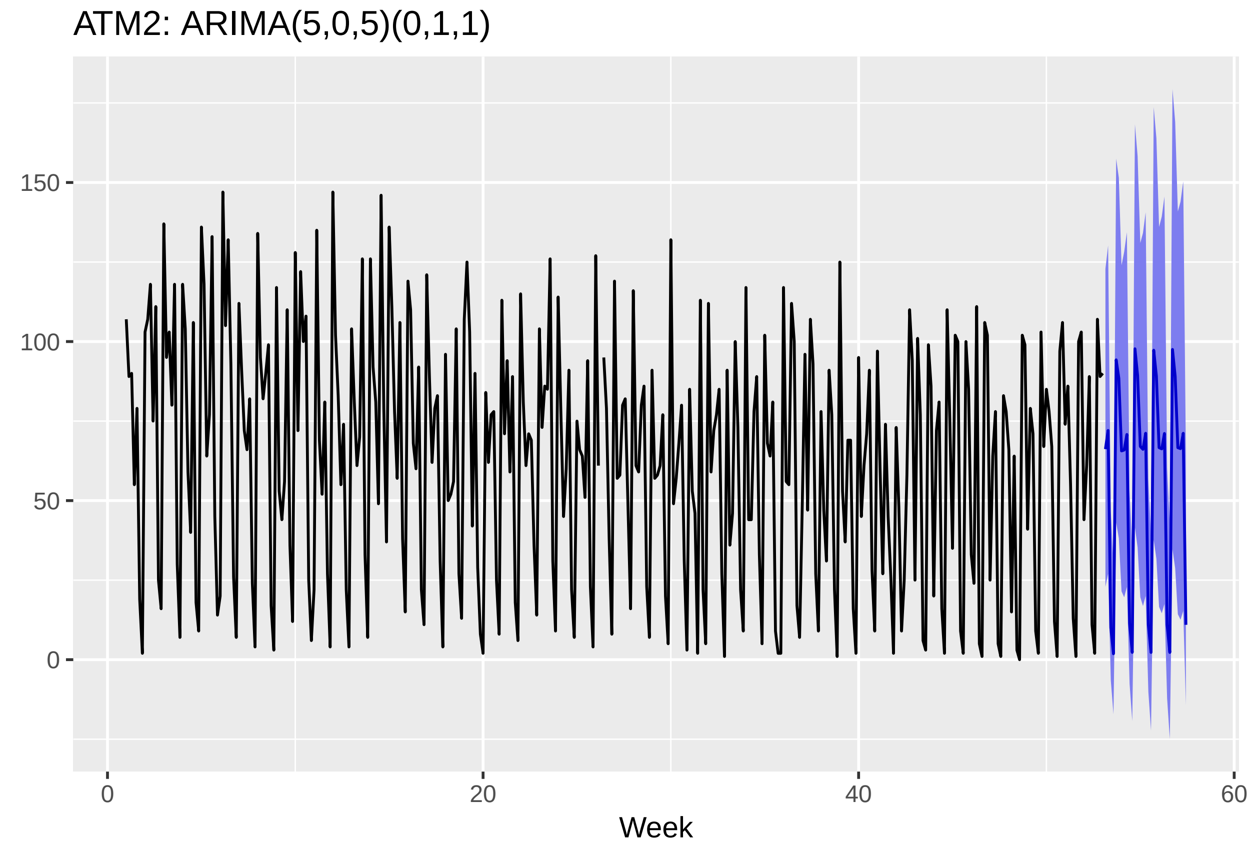
```r
ggtsdisplay(resid(fit_atm2), points = FALSE, plot.type = "histogram",

            main = "Residuals for ARIMA(5,0,5)(0,1,1) of ATM2 withdrawals",

            xlab = "Week", ylab = "Residual")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```



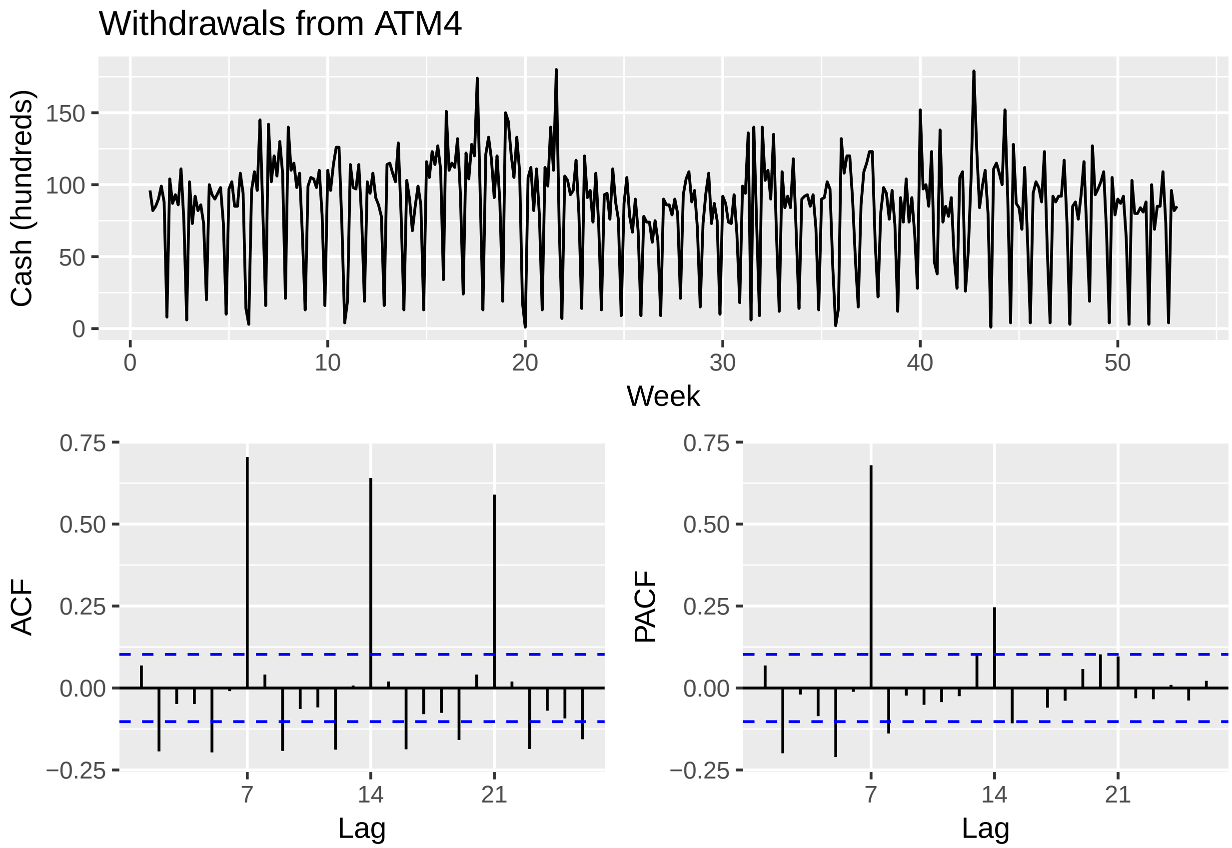
P-value is greater than 0.05 and the residual appear to be normally distributed with a mean of 0. It can be used for forecast ATM2.

```
forecast_atm2<- forecast(fit_atm2, 31, level = 95)
autoplot(forecast_atm2) +

    labs(title = "ATM2: ARIMA(5,0,5)(0,1,1)", x = "Week", y = NULL) +

    theme(legend.position = "none")
```



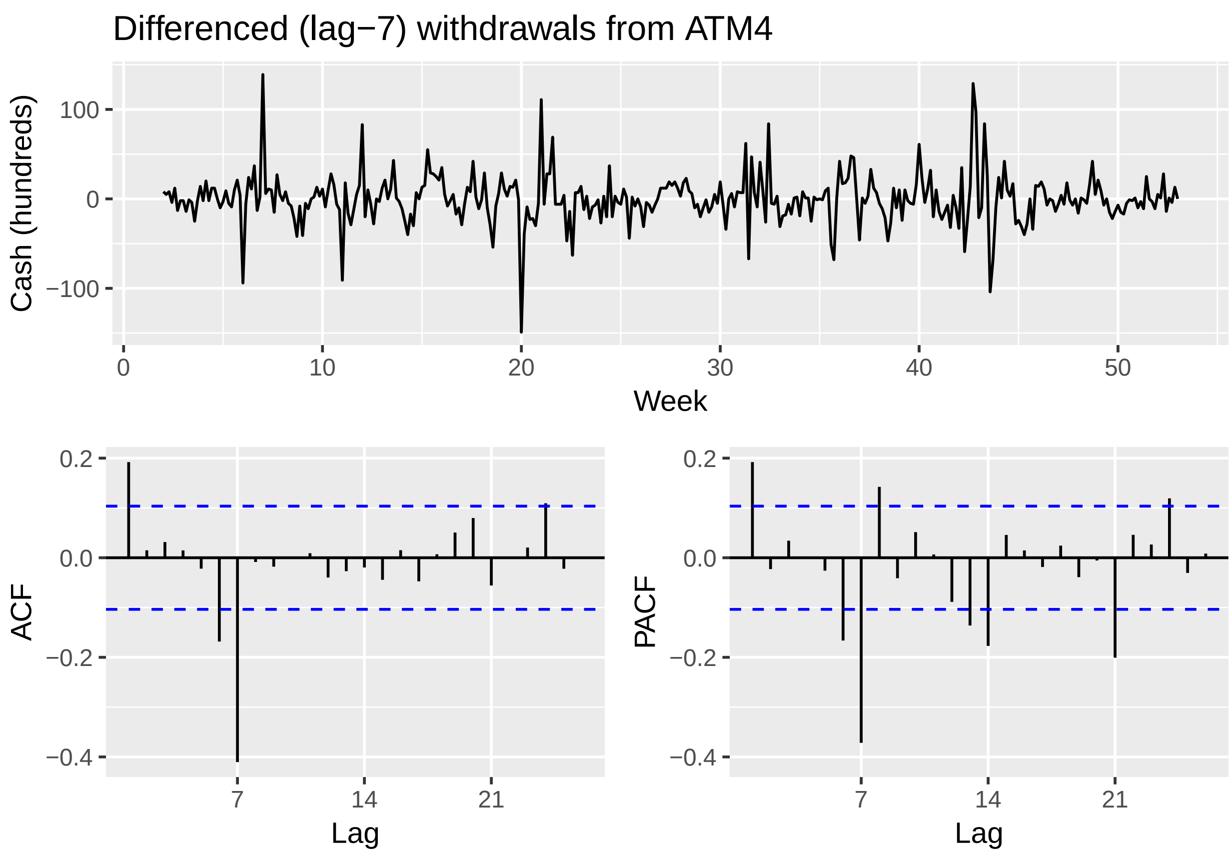
##ATM4 ATM4 has the same seasonality as ATM1 and ATM2. We will use the previous step to evaluate ATM4 model.

```
#Minimze the effect of the big withdraw in the day by using the median of the ATM4 dataset

atm4_freq[which.max(atm4_freq)] <- median(atm4_freq, na.rm = TRUE)

ggtsdisplay(atm4_freq, points = FALSE,

            main = "Withdrawals from ATM4", xlab = "Week", ylab = "Cash (hundreds)")
```



We notice a difference lag of 7.

```
ggtsdisplay(diff(atm4_freq, 7), points = FALSE,

            main = "Differenced (lag-7) withdrawals from ATM4",

            xlab = "Week", ylab = "Cash (hundreds)")
```



ARIMA model for ATM4 will be evaluated.

```
# get optimal lambda for Box-cox transformation

lambda_atm4 <- BoxCox.lambda(atm4_freq)

aic_atm(0,2,5,0,2,5)
```

```
## [1] 2365.837
```

```
# create possible combinations of p, q, P, Q except all zero

expand.grid(p = c(0, 2, 5), q = c(0, 2, 5), P = 0:1, Q = 0:1) %>%

  filter(p > 0 | q > 0 | P > 0 | Q > 0) %>%

  # calculate AIC for models

  mutate(aic = pmap_dbl(list(p, 0, q, P, 1, Q), aic_atm)) %>%

  # return minimum AIC

  slice(which.min(aic))
```

```
##   p q P Q      aic
## 1 2 2 0 1 2323.517
```

Let's fit the ARIMA model with the values (0,0,2)(0,1,1)

```r
fit_atm4<-Arima(atm4_freq, order = c(0, 0, 2), seasonal = c(0, 1, 1), lambda = lambda_atm4)
summary(fit_atm4)
```

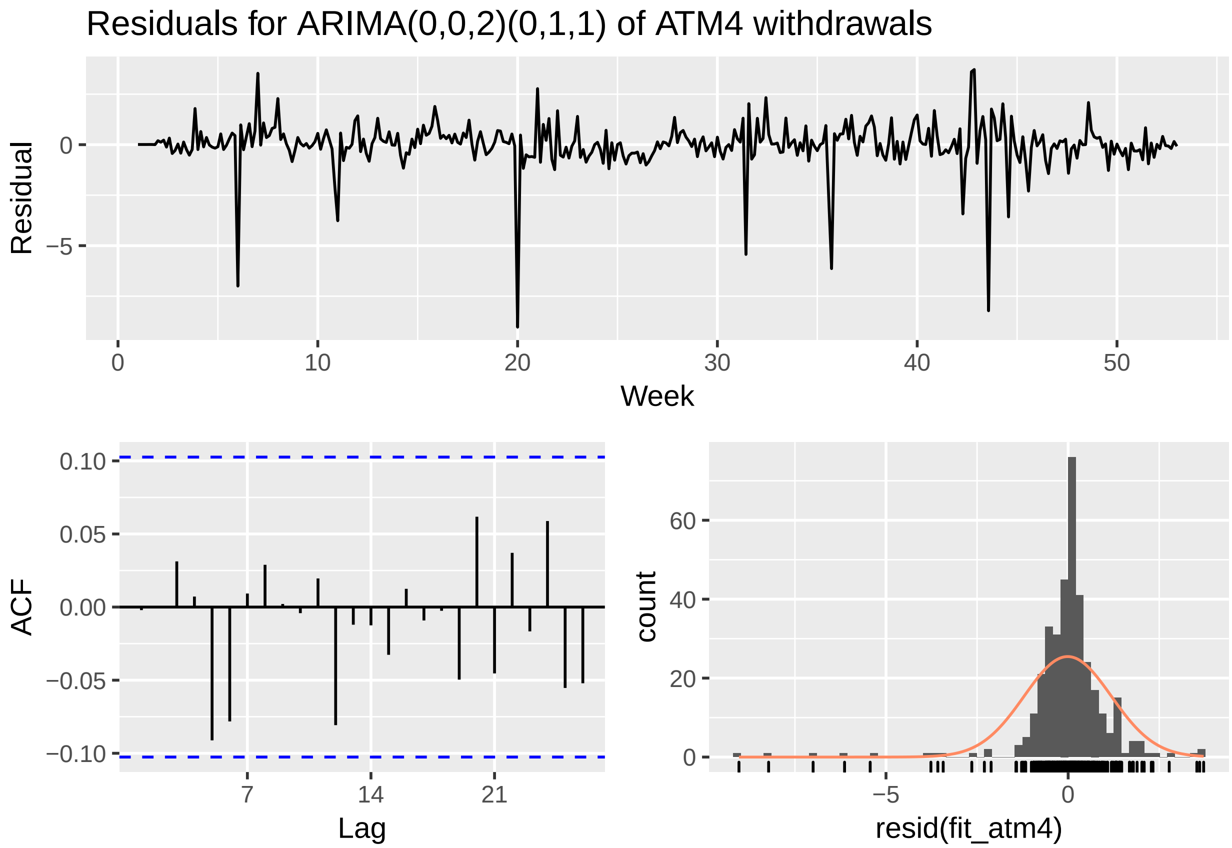
```
## Series: atm4_freq
## ARIMA(0,0,2)(0,1,1)[7]
## Box Cox transformation: lambda= 0.2355973
##
## Coefficients:
##          ma1      ma2     sma1
##       0.1094  -0.1089  -0.6468
## s.e.  0.0524   0.0523   0.0422
##
## sigma^2 estimated as 1.467:  log likelihood=-576.96
## AIC=1161.92   AICc=1162.03   BIC=1177.44
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 2.356651 24.88094 15.90136 -85.71176 104.5953 0.9023123
##                    ACF1
## Training set 0.02127326
```

Let's investigate the residuals using Ljung-box test

```r
Box.test(resid(fit_atm4), type = "L", fitdf = 3, lag = 7)
```

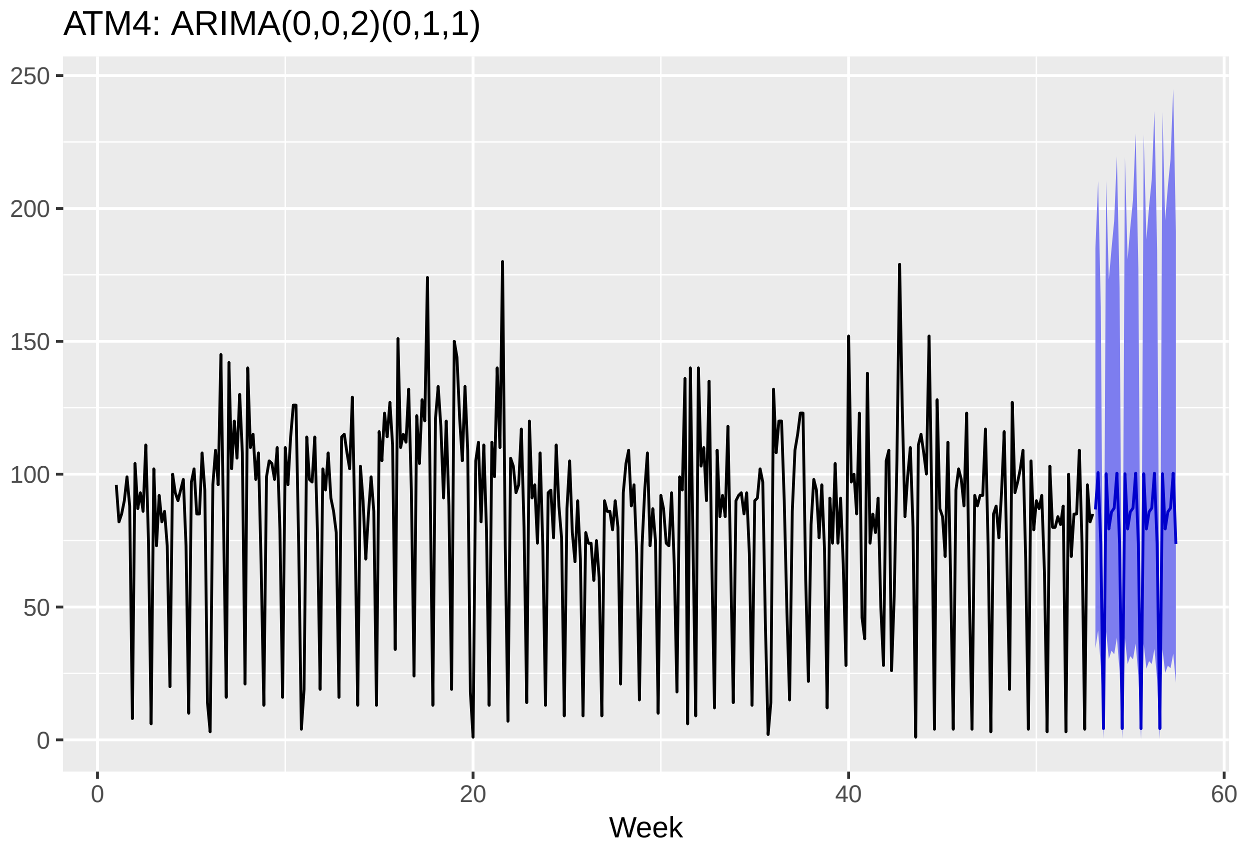
```
##
##  Box-Ljung test
##
## data:  resid(fit_atm4)
## X-squared = 5.7899, df = 4, p-value = 0.2154
```

```r
ggtsdisplay(resid(fit_atm4), points = FALSE, plot.type = "histogram",

            main = "Residuals for ARIMA(0,0,2)(0,1,1) of ATM4 withdrawals",

            xlab = "Week", ylab = "Residual")
```



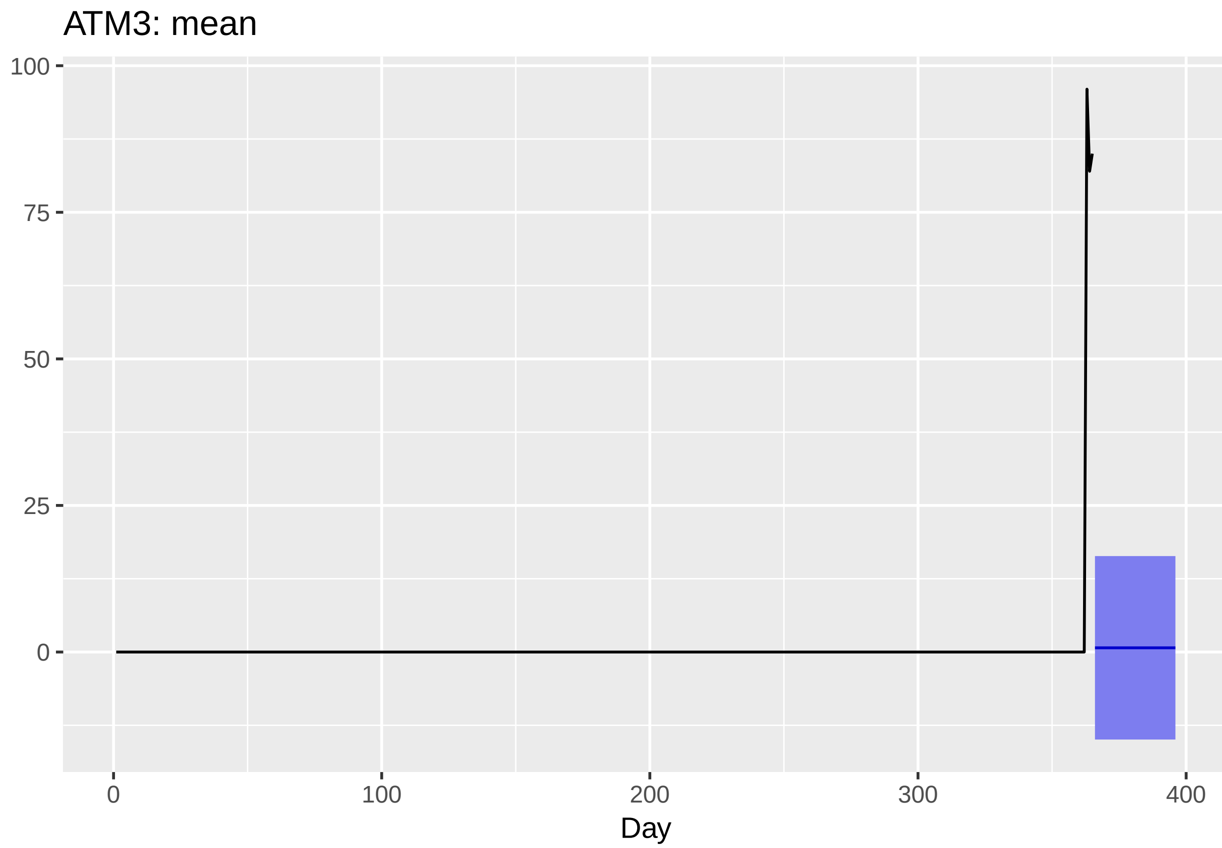
It is normally distributed around a mean of 0.p-value is also greater than 0.05. We can use the model to forecast.

```
forecast_atm4<- forecast(fit_atm4, 31, level = 95)
autoplot(forecast_atm4) +

    labs(title = "ATM4: ARIMA(0,0,2)(0,1,1)", x = "Week", y = NULL) +

    theme(legend.position = "none")
```



##ATM3 Since ATM3 contains limited data we will use the mean forecast method.

```
forecast_atm3 <- meanf(atm3, 31, level = 95)
autoplot(forecast_atm3) +

    labs(title = "ATM3: mean", x = "Day", y = NULL) +

    theme(legend.position = "none")
```



Writing the forecast to a CSV file

```
data_frame(DATE = rep(max(atm$DATE) + 1:31, 4),
           atm = rep(names(atm)[-1], each = 31),
           Cash = c(forecast_atm1$mean, forecast_atm2$mean,
                    forecast_atm3$mean, forecast_atm4$mean)) %>%
  write_csv("C:/Users/hangr/Documents/fall2019/Data624/project1_forecast_atm.csv")
```