Nicholas Larson & Robert Smith

Time Series Analysis Project

November 26, 2013

Background

```
bt <- read.table("Bath Tissue Movement.csv")
bt <- ts(bt, frequency = 7)

adf.test(bt)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  bt
## Dickey-Fuller = -4.381, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
pp.test(bt)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  bt
## Dickey-Fuller Z(alpha) = -156.9, Truncation lag parameter = 5,
## p-value = 0.01
## alternative hypothesis: stationary
```

```
bt_diff <- diff(bt, 1, 7)

adf.test(bt_diff)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  bt_diff
## Dickey-Fuller = -20.28, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

```
pp.test(bt_diff)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  bt_diff
## Dickey-Fuller Z(alpha) = -676.1, Truncation lag parameter = 5,
## p-value = 0.01
## alternative hypothesis: stationary
```

Plots

```
layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
plot(bt)
acf(bt)
pacf(bt)
```

Auto ARIMA

```
(bt_fcst <- auto.arima(bt, d = 1, D = 1))
```

```
## Series: bt
## ARIMA(0,1,2)(0,1,2)[7]
##
## Coefficients:
##          ma1     ma2    sma1   sma2
##       -0.635  -0.198  -1.094  0.137
## s.e.   0.049   0.053   0.054  0.055
##
## sigma^2 estimated as 8835:  log likelihood=-2347
## AIC=4705   AICc=4705   BIC=4725
```

```
plot(forecast(bt_fcst))
```

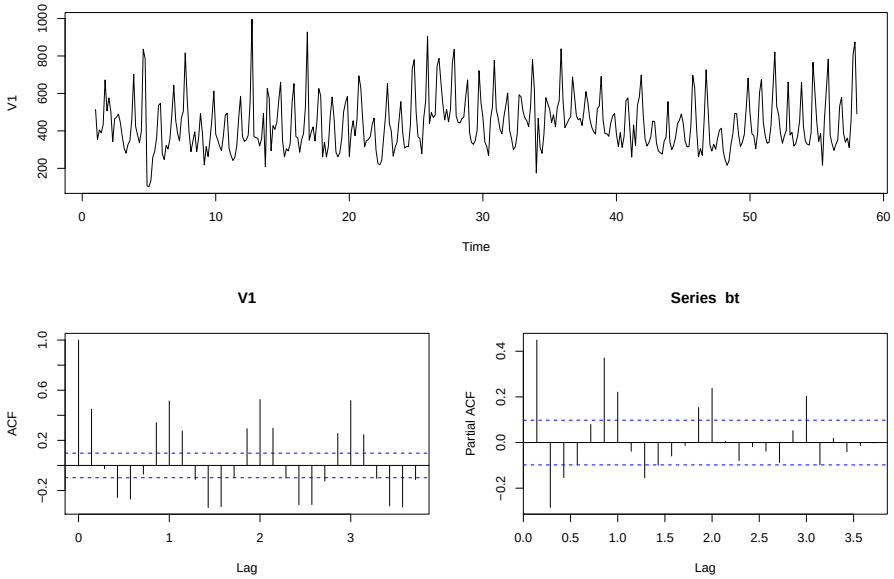

Figure 1: plot of chunk Plots

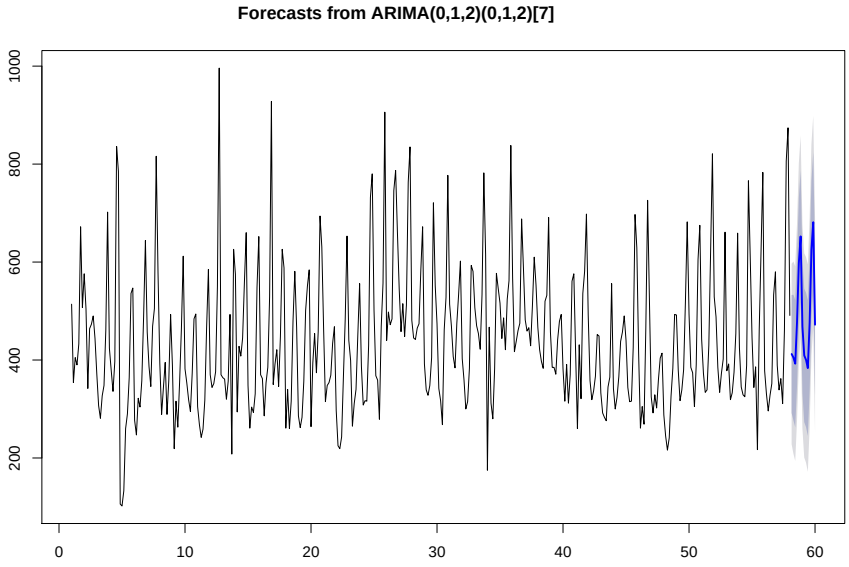

Figure 2: plot of chunk AutoArima

Validation

For the Model validation we decided to simulate with both Monte Carlo and Bootstrap estimates. Given the model we fit above,

```
mc.estimate <- function(TS, Order, Seasonal, reps) {
    library(fGarch)  # for skew-t distribution
    library(forecast)  #for simulate.Arima function
    stopifnot(length(reps) == 1)

    Arima_est <- Arima(TS, order = Order, seasonal = Seasonal)

    for (i in 1:reps) {
        X <- simulate(Arima_est, bootstrap = TRUE)
        Est <- Arima(X, order = Order, seasonal = Seasonal)
        if (i == 1) {
            Vars <- MLE <- matrix(0, reps, length(Est$coef))
        }
        MLE[i, ] <- Est$coef
        Vars[i, ] <- diag(Est$var.coef)
    }

    df <- as.data.frame(cbind(MLE, Vars))
    colnames(df) <- c(paste0("MLE", names(Est$coef)), paste0("VAR",
        names(diag(Est$var.coef))))

    return(df)
}

bootstrap.estimate <- function(TS, Order, Seasonal, reps) {
    library(fGarch)  # for skew-t distribution
    library(forecast)  #for simulate.Arima function
    stopifnot(length(reps) == 1)

    Arima_est <- Arima(TS, order = Order, seasonal = Seasonal)
    res <- Arima_est$residuals
    mle <- Arima_est$coef

    for (i in 1:reps) {
        et <- sample(res, length(res), replace = TRUE)
        est2 <- simulate(Arima_est, innov = et)
        est3 <- Arima(est2, order = Order, seasonal = Seasonal)

        X <- simulate(Arima_est, bootstrap = TRUE)
```

```
        Est <- Arima(X, order = Order, seasonal = Seasonal)

        if (i == 1) {
            Vars <- MLE <- matrix(0, reps, length(Est$coef))
        }
        MLE[i, ] <- Est$coef
        Vars[i, ] <- diag(Est$var.coef)
    }

    df <- as.data.frame(cbind(MLE, Vars))
    colnames(df) <- c(paste0("MLE", names(Est$coef)), paste0("VAR",
        names(diag(Est$var.coef))))

    return(df)
}

Order <- 0:2
Seasonal <- list(order = Order, period = 7)

mc <- mc.estimate(bt, Order, Seasonal, 100)

boot <- bootstrap.estimate(bt, Order, Seasonal, 100)


p1 <- ggplot(mc, aes(x = MLEma1)) + geom_histogram(aes(y = ..density..),
    binwidth = 0.01) + stat_density(col = "red", geom = "line") +
    geom_vline(xintercept = bt_fcst$coef[1], color = "blue") +
    geom_vline(xintercept = mean(mc[[1]]), color = "green") +
    labs(title = "Monte Carlo MLE MA(1)", x = "MLE MA(1)")

p2 <- ggplot(mc, aes(x = MLEma2)) + geom_histogram(aes(y = ..density..),
    binwidth = 0.01) + stat_density(col = "red", geom = "line") +
    geom_vline(xintercept = bt_fcst$coef[2], color = "blue") +
    geom_vline(xintercept = mean(mc[[2]]), color = "green") +
    labs(title = "Monte Carlo MLE MA(2)", x = "MLE MA(2)")

p3 <- ggplot(mc, aes(x = MLEsma1)) + geom_histogram(aes(y = ..density..),
    binwidth = 0.01) + stat_density(col = "red", geom = "line") +
    geom_vline(xintercept = bt_fcst$coef[3], color = "blue") +
    geom_vline(xintercept = mean(mc[[3]]), color = "green") +
    labs(title = "Monte Carlo MLE SMA(1)", x = "MLE SMA(1)")

p4 <- ggplot(mc, aes(x = MLEsma2)) + geom_histogram(aes(y = ..density..),
    binwidth = 0.01) + stat_density(col = "red", geom = "line") +
    geom_vline(xintercept = bt_fcst$coef[4], color = "blue") +
    geom_vline(xintercept = mean(mc[[4]]), color = "green") +
```

```
    labs(title = "Monte Carlo MLE SMA(2)", x = "MLE SMA(2)")

grid.arrange(p1, p2, p3, p4, nrow = 2)

## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```

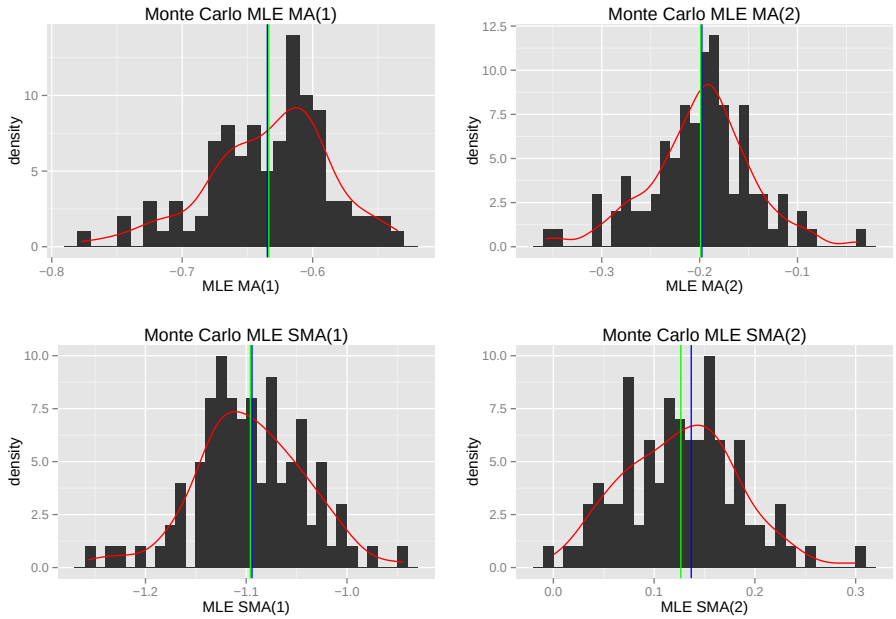

Figure 3: plot of chunk validation

```
p1 <- ggplot(boot, aes(x = MLEma1)) + geom_histogram(aes(y = ..density..),
    binwidth = 0.01) + stat_density(col = "red", geom = "line") +
    geom_vline(xintercept = bt_fcst$coef[1], color = "blue") +
    geom_vline(xintercept = mean(boot[[1]]), color = "green") +
    labs(title = "Boostrap MLE MA(1)", x = "MLE MA(1)")

p2 <- ggplot(boot, aes(x = MLEma2)) + geom_histogram(aes(y = ..density..),
    binwidth = 0.01) + stat_density(col = "red", geom = "line") +
    geom_vline(xintercept = bt_fcst$coef[2], color = "blue") +
    geom_vline(xintercept = mean(boot[[2]]), color = "green") +
```

```
    labs(title = "Boostrap MLE MA(2)", x = "MLE MA(2)")

p3 <- ggplot(boot, aes(x = MLEsma1)) + geom_histogram(aes(y = ..density..),
    binwidth = 0.01) + stat_density(col = "red", geom = "line") +
    geom_vline(xintercept = bt_fcst$coef[3], color = "blue") +
    geom_vline(xintercept = mean(boot[[3]]), color = "green") +
    labs(title = "Boostrap MLE SMA(1)", x = "MLE SMA(1)")

p4 <- ggplot(boot, aes(x = MLEsma2)) + geom_histogram(aes(y = ..density..),
    binwidth = 0.01) + stat_density(col = "red", geom = "line") +
    geom_vline(xintercept = bt_fcst$coef[4], color = "blue") +
    geom_vline(xintercept = mean(boot[[4]]), color = "green") +
    labs(title = "Boostrap MLE SMA(2)", x = "MLE SMA(2)")

grid.arrange(p1, p2, p3, p4, nrow = 2)

## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```

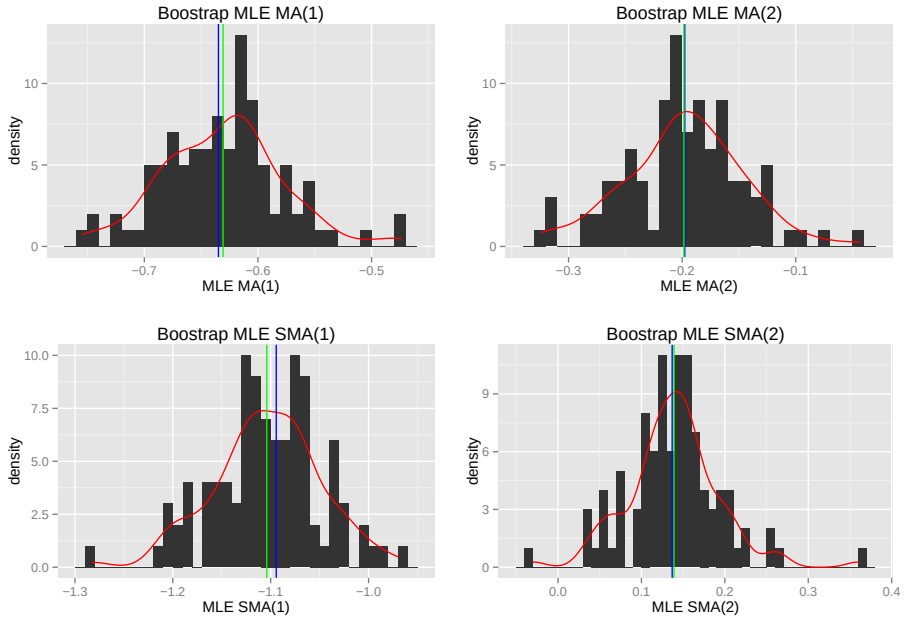

Figure 4: plot of chunk validation