HW 4: Fitting Accident Data with regression line

Robert Smith

November 19, 2013

1. Read in acci.txt file from the course web site. Plot the time series. Take difference with lag 12. Plot the differenced data. Call it D2.

```
acci <- read.table("acci.txt", header = TRUE)
D1 <- ts(acci, frequency = 12)
D2 <- diff(D1, d = 12)

##
##  Augmented Dickey-Fuller Test
##
## data:  D2
## Dickey-Fuller = -19.95, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary

pp.test(D2)

##
##  Phillips-Perron Unit Root Test
##
## data:  D2
## Dickey-Fuller Z(alpha) = -105.5, Truncation lag parameter = 3,
## p-value = 0.01
## alternative hypothesis: stationary

kpss.test(D2)

##
##  KPSS Test for Level Stationarity
##
## data:  D2
## KPSS Level = 0.0483, Truncation lag parameter = 1, p-value = 0.1


layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))
```

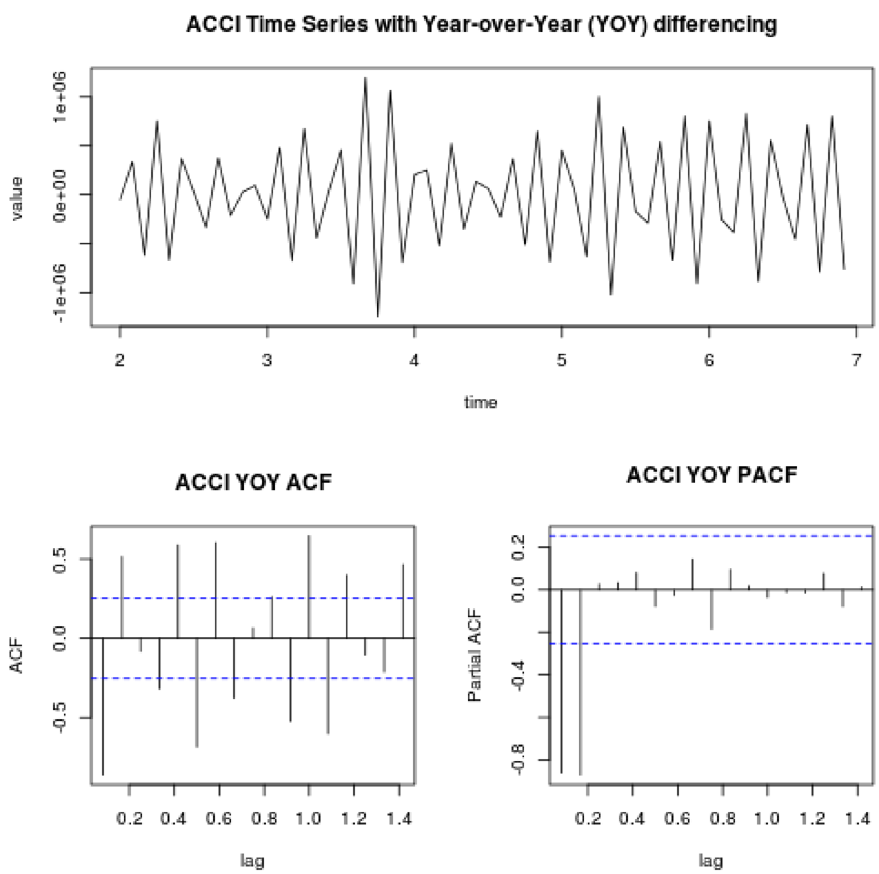

Figure 1: plot of chunk q1

2. Fit D2 with linear trend using OLS. Is the slope significant? How did you determine? Can the output from the screen be trusted? Based on the RSE, adjusted ρ^2 and p-value we fail to reject H_0 and cannot regard the slope of the OLS fit of the year-over-year differences as signficant. This indicates that the values obtained in D2 are dependent upon time and should probably not be trusted. When we plot the forecast of the residuals we see a repeating pattern in the data and the forecast as well.

```
time <- 1:length(D2)
(err_mod <- summary(lm_D2 <- lm(D2 ~ time)))
```

```
##
## Call:
## lm(formula = D2 ~ time)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1245097  -504589    21005   486720  1194118
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    14854     159786    0.09     0.93
## time            -663       4556   -0.15     0.88
##
## Residual standard error: 611000 on 58 degrees of freedom
## Multiple R-squared:  0.000365,   Adjusted R-squared:  -0.0169
## F-statistic: 0.0212 on 1 and 58 DF,  p-value: 0.885
```

```
resid <- lm_D2$residuals

(arima_mdl <- auto.arima(ts(resid, start = c(2, 1), freq = 12)))
```

```
## Series: ts(resid, start = c(2, 1), freq = 12)
## ARIMA(4,0,0)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1     ar2     ar3     ar4    sma1   drift
##       -2.965  -3.879  -2.629  -0.779  -0.397  659.77
## s.e.   0.087   0.204   0.202   0.083   0.224   28.18
##
## sigma^2 estimated as 1.06e+09:  log likelihood=-425.1
## AIC=864.2   AICc=867   BIC=877.4
```

```
adf.test(resid)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  resid
## Dickey-Fuller = -19.95, Lag order = 3, p-value = 0.01
## alternative hypothesis: stationary
```

```r
pp.test(resid)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  resid
## Dickey-Fuller Z(alpha) = -105.5, Truncation lag parameter = 3,
## p-value = 0.01
## alternative hypothesis: stationary
```

```r
layout(matrix(c(1, 2, 3), 3, 1, byrow = TRUE))
plot(D2)
abline(lm_D2)
plot(x = 1:length(resid), y = resid, type = "l", main = "Fits vs. Residuals",
    ylab = "residuals")
plot(forecast(arima_mdl))
```

```r
layout(matrix(c(1, 2, 3, 4), 2, 2, byrow = TRUE))
plot(lm_D2)
```

3. Fit residuals from (2) with seasonal ARIMA with d=0, D=0. (i.e. ARIMA(p,0,q)x(P,0,Q)12). Is the seasonal part necessary? Based on the results above, the standard error for the seasonal component is significant and therefore should be included in the forecast.

```r
(arima00 <- auto.arima(ts(lm_D2$residuals, start = c(2, 1), freq = 12), d = 0,
    D = 0))
```

```
## Series: ts(lm_D2$residuals, start = c(2, 1), freq = 12)
## ARIMA(3,0,0)(1,0,0)[12] with zero mean
##
## Coefficients:
```

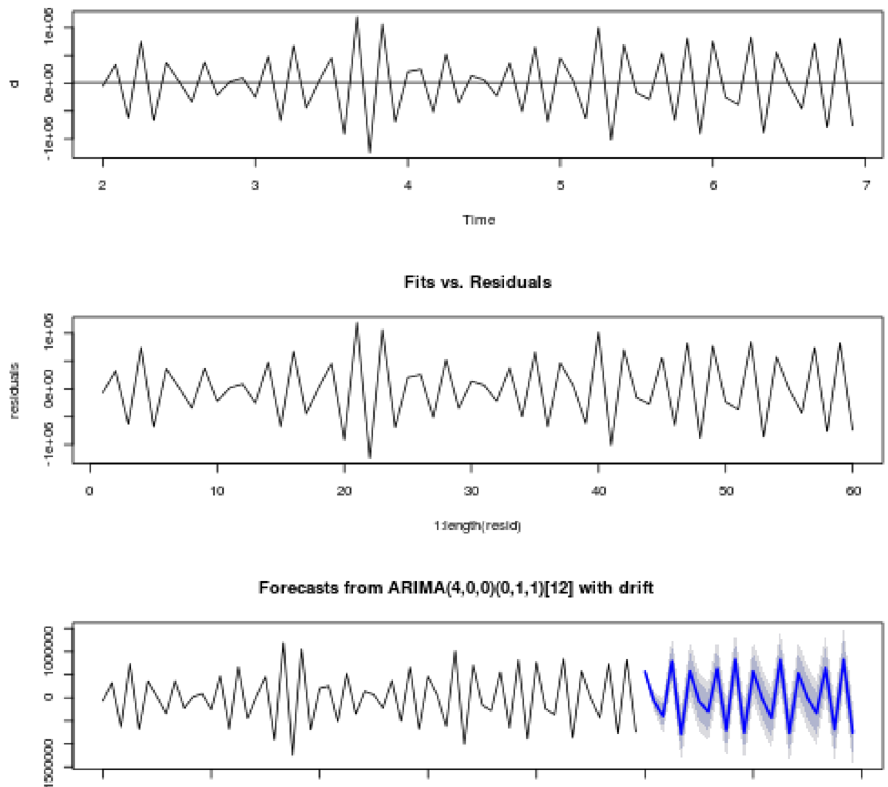

Figure 2: plot of chunk q2

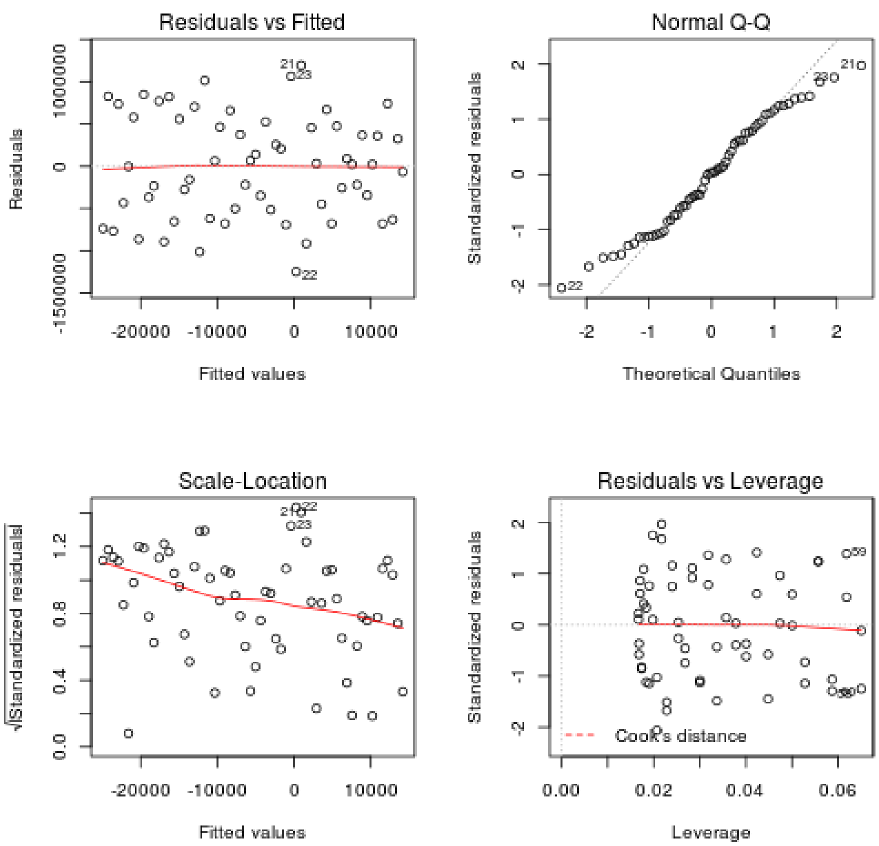

Figure 3: plot of chunk q2

```
##           ar1      ar2      ar3    sar1
##        -2.168   -1.869   -0.623   0.782
## s.e.    0.096    0.170    0.100   0.081
##
## sigma^2 estimated as 4.82e+09:  log likelihood=-762.4
## AIC=1535   AICc=1536   BIC=1545
```

4. Using the best model from #3, predict twelve months ahead in D2.

```
plot(forecast(arima00, h = 12))
```

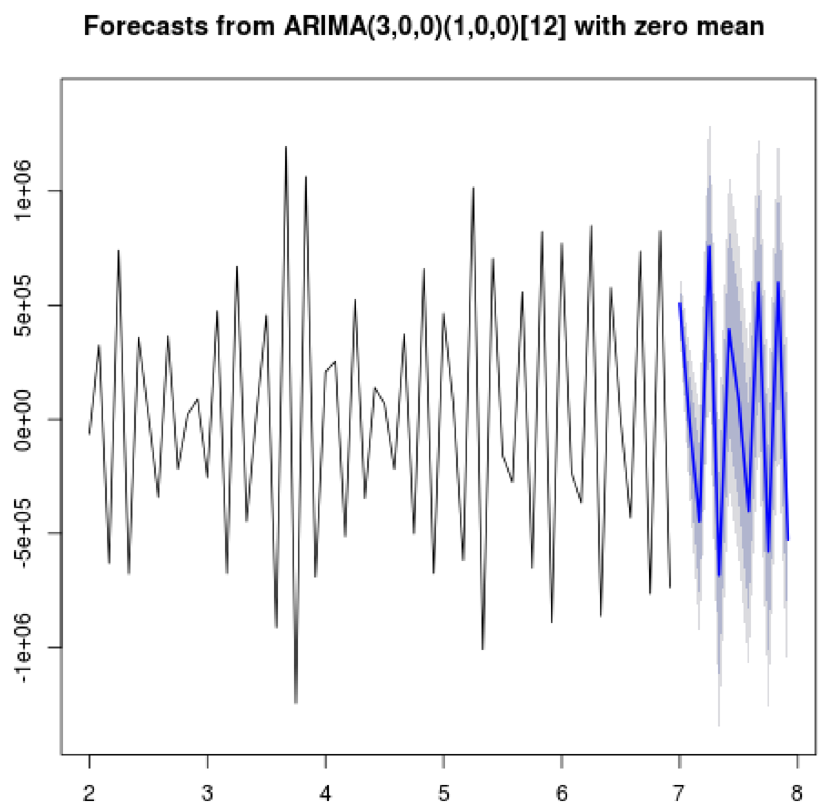

Figure 4: plot of chunk q4

5. Using the prediction from #4, predict twelve months ahead in D1 (original TS).

```
(q5D1 <- Arima(D1, order = c(3, 0, 0), seasonal = list(order = c(1, 0, 0), period = 12)))
```

```
## Series: D1
## ARIMA(3,0,0)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##         ar1    ar2    ar3   sar1  intercept
##       0.644  0.109  0.050  0.876     9317.0
## s.e.  0.124  0.140  0.126  0.049      915.4
##
## sigma^2 estimated as 121442:  log likelihood=-532.9
## AIC=1078   AICc=1079   BIC=1092
```

```
q5 <- forecast(q5D1)
plot(q5)
```

6. Using your model from In-class Ex2-#2, (ARIMA(p,1,q)x(P,1,Q)12 model), predict twelve months ahead in D1.

```
(q6D1 <- auto.arima(D1, d = 1, D = 1))
```

```
## Series: D1
## ARIMA(0,1,1)(0,1,1)[12]
##
## Coefficients:
##          ma1    sma1
##       -0.426  -0.558
## s.e.   0.123   0.179
##
## sigma^2 estimated as 99480:  log likelihood=-425.5
## AIC=857.1   AICc=857.5   BIC=863.3
```

```
q6 <- forecast(q6D1)
plot(q6)
```

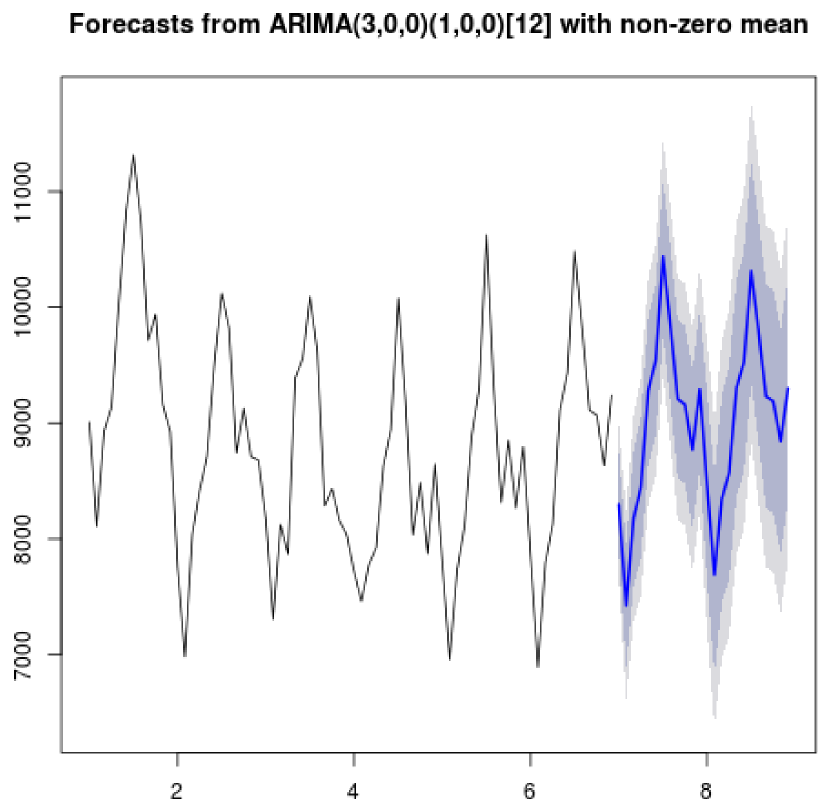

Figure 5: plot of chunk q5

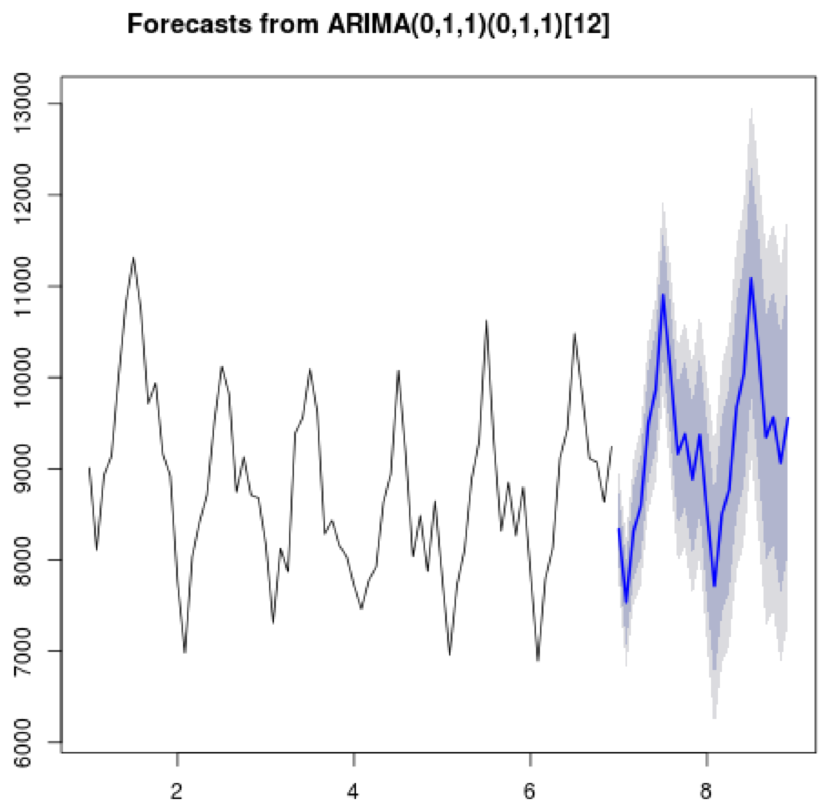

Figure 6: plot of chunk q6

7. Using your model from In-class Ex2-#3, (ARIMA(p,0,q)x(P,1,Q)12 model), predict twelve months ahead in D1.

```r
(q7D1 <- auto.arima(D1, d = 0, D = 1))
```

```
## Series: D1
## ARIMA(2,0,0)(1,1,0)[12] with drift
##
## Coefficients:
##         ar1    ar2    sar1   drift
##       0.592  0.258  -0.350  -13.71
## s.e.  0.138  0.143   0.142   18.56
##
## sigma^2 estimated as 138798:  log likelihood=-348
## AIC=705.9   AICc=707   BIC=716.4
```

```r
q7 <- forecast(q7D1)
plot(q7)
```

8. Compare your prediction from #5, #6, and #7. Plot the one-month predictions on the sampe plot. Which one do you like the best? Based on the results from questions 5, 6 & 7 I believe the model fit in question #7 to be the best based on its AIC, AICc & BIC statistics being the smallest for the three questions and for the standard errors it produces being among the smallest of the group.

```r
plot(D1, xlim = c(1, 8))
points(x = seq(7, 8, length.out = 12), y = q7$mean[1:12])
points(x = seq(7, 8, length.out = 12), y = q6$mean[1:12], col = "red")
points(x = seq(7, 8, length.out = 12), y = q5$mean[1:12], col = "blue")
```

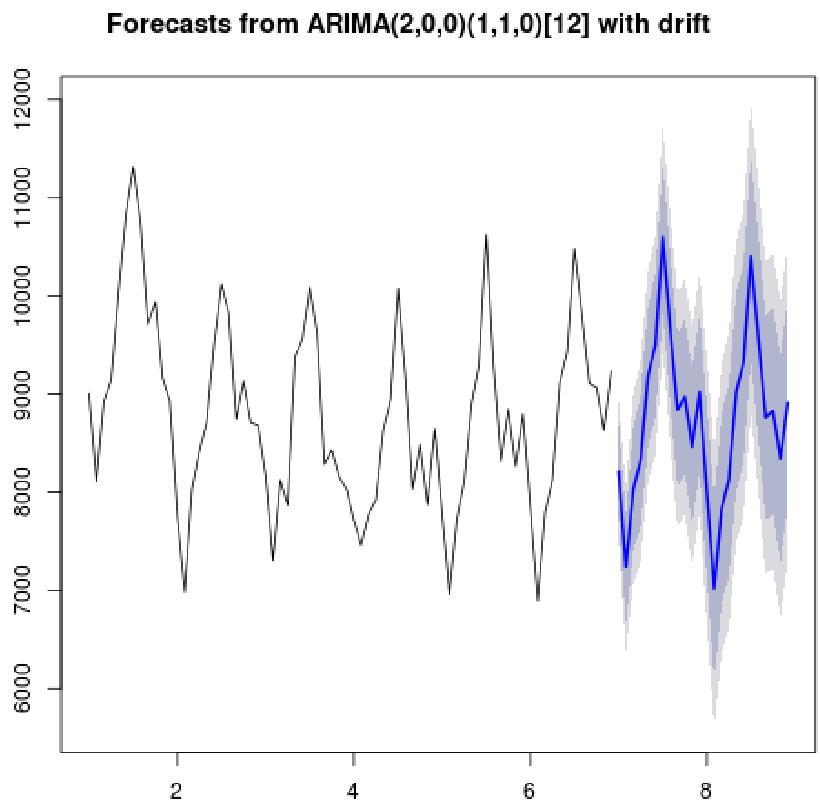

Figure 7: plot of chunk q7

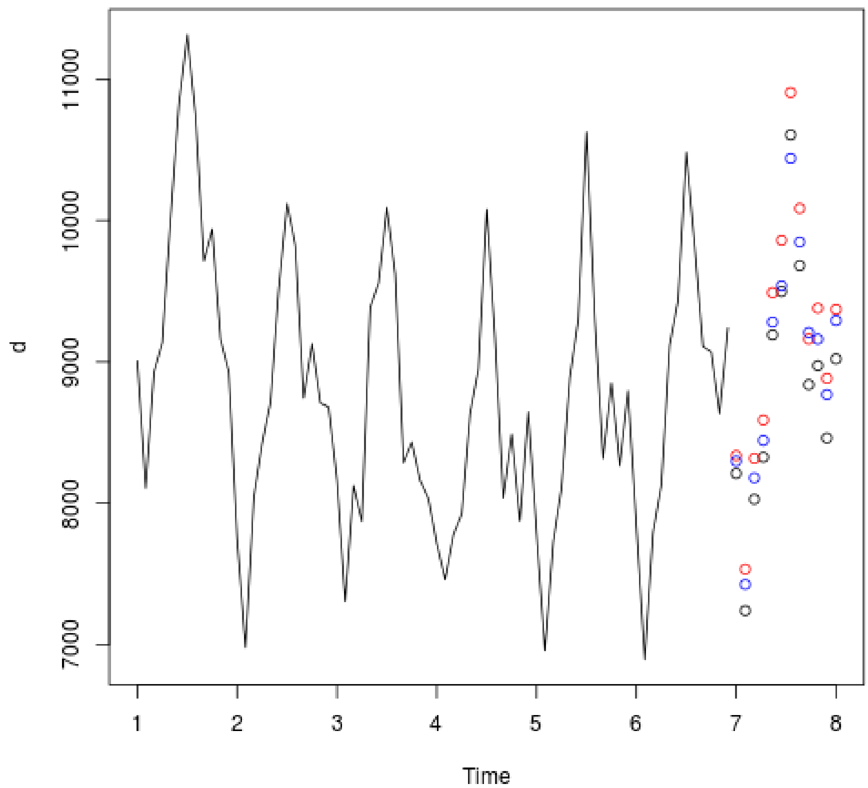

Figure 8: plot of chunk q8