Volatility Clustering

Ashley (Qingyang) Mu

11/02/2020

Read the document

```
dat <- read.table(file = '/Users/Qingyang/Downloads/Week5_Test_Sample.csv',header=TRUE)
head(dat)
```

```
##       Output      Input
## 1 -0.3056753 -0.5937954
## 2 -2.1055764 -2.0317675
## 3 -1.7088101 -2.1446915
## 4 -1.0717098 -1.5771034
## 5  1.6458175  0.2758625
## 6 -2.5361552 -1.7552214
```

Fit the general model

```
GeneralModel <- lm(dat$Output~dat$Input,data = dat)
summary(GeneralModel)
```

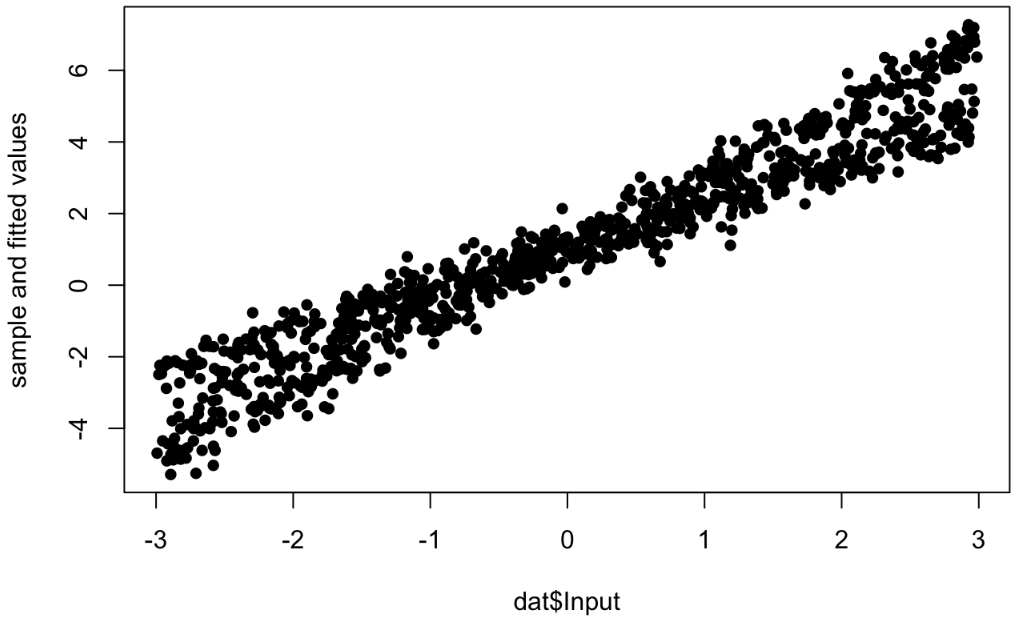
```
##
## Call:
## lm(formula = dat$Output ~ dat$Input, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.04901 -0.54000 -0.01178  0.56074  1.79382
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.00072    0.02334   42.88   <2e-16 ***
## dat$Input    1.55319    0.01353  114.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7381 on 998 degrees of freedom
## Multiple R-squared:  0.9296, Adjusted R-squared:  0.9295
## F-statistic: 1.317e+04 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
GeneralModel$coefficients
```

```
## (Intercept)   dat$Input
##    1.000719    1.553193
```

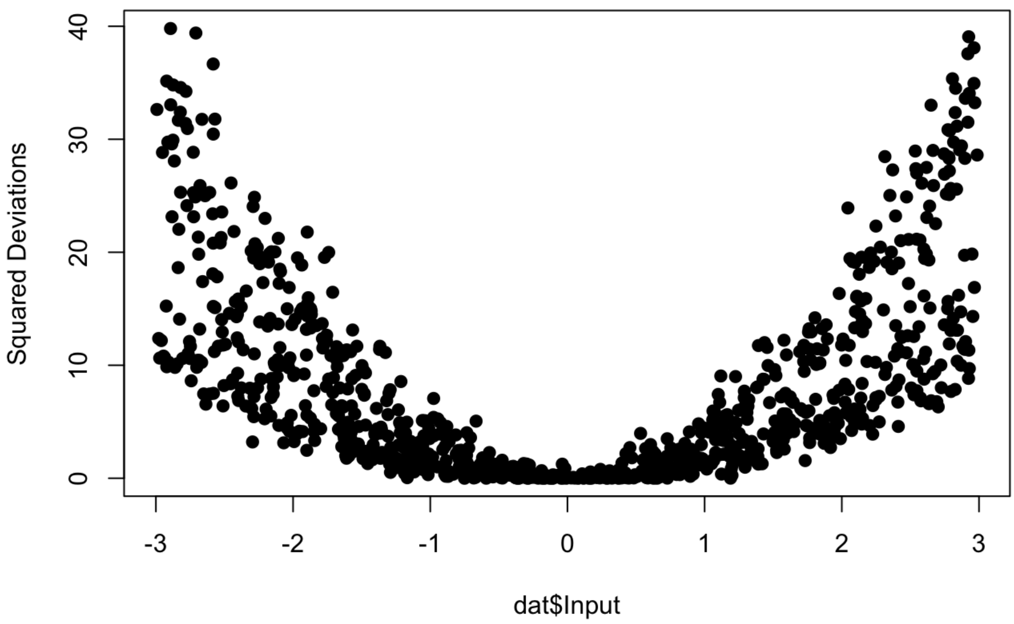
Plot the data and check potential clustering opportunities

```
matplot(dat$Input,dat$Output,type='p',pch=16,ylab = 'sample and fitted values')
```



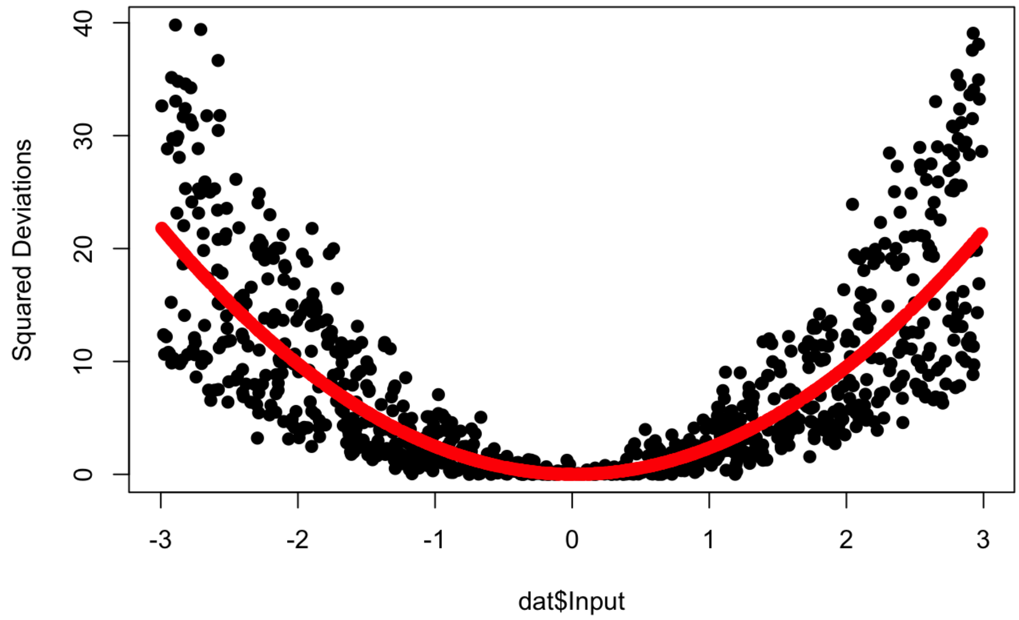
Make a plot of squared deviations zi=(yi−y¯)^2.

```
plot(dat$Input,(dat$Output-mean(dat$Output))^2, type="p",pch=19,
     ylab="Squared Deviations")
```



Find the parabola corresponding to fitted mode

```
plot(dat$Input,(dat$Output-mean(dat$Output))^2, type="p",pch=19,
     ylab="Squared Deviations")
clusteringParabola <- (GeneralModel$fitted.values-mean(dat$Output))^2
points(dat$Input,clusteringParabola,pch=19,col="red")
```



Define the separating sequence Unscrambling.Sequence.Steeper.var, such that it is equal to TRUE for steeper slope subsample and FALSE for flatter slope subsample.

```
Unscrambling.Sequence.Steeper.var <- (dat$Output-mean(dat$Output))^2>=(GeneralModel$fitted.values-mean(dat$O
utput))^2
head(Unscrambling.Sequence.Steeper.var)
```

```
##     1     2     3     4     5     6
##  TRUE FALSE FALSE FALSE  TRUE  TRUE
```

Separate the sample into steeper and flatter part. Create data frames. Define two subsamples with NAs in the Output columns

```
nSample <- length(dat$Input)
Subsample.Steeper.var<-data.frame(steeperInput.var=dat$Input,steeperOutput.var=rep(NA,nSample))
Subsample.Flatter.var<-data.frame(flatterInput.var=dat$Input,flatterOutput.var=rep(NA,nSample))
```

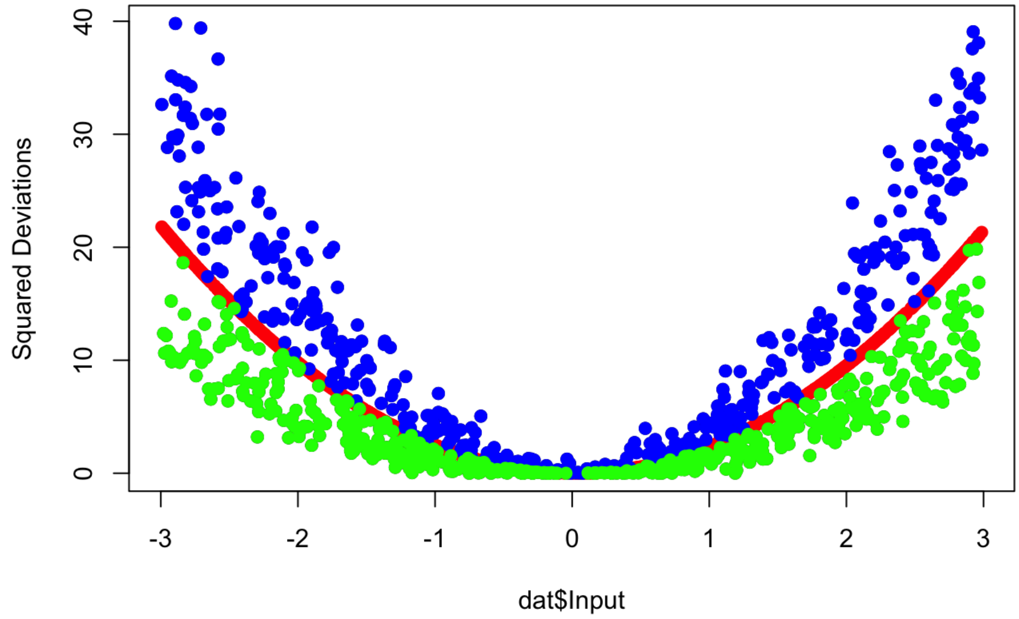
Fill in the unscrambled outputs instead of NAs where necessary

```
Subsample.Steeper.var[Unscrambling.Sequence.Steeper.var,2]<-
  dat[Unscrambling.Sequence.Steeper.var,1]
Subsample.Flatter.var[!Unscrambling.Sequence.Steeper.var,2]<-
  dat[!Unscrambling.Sequence.Steeper.var,1]
head(cbind(dat,Subsample.Steeper.var,Subsample.Flatter.var),10)
```

```
##         Output      Input steeperInput.var steeperOutput.var flatterInput.var
## 1  -0.3056753 -0.5937954       -0.5937954        -0.3056753       -0.5937954
## 2  -2.1055764 -2.0317675       -2.0317675                NA       -2.0317675
## 3  -1.7088101 -2.1446915       -2.1446915                NA       -2.1446915
## 4  -1.0717098 -1.5771034       -1.5771034                NA       -1.5771034
## 5   1.6458175  0.2758625        0.2758625         1.6458175        0.2758625
## 6  -2.5361552 -1.7552214       -1.7552214        -2.5361552       -1.7552214
## 7   3.3940221  1.1649775        1.1649775         3.3940221        1.1649775
## 8   1.2448116  0.4300578        0.4300578                NA        0.4300578
## 9  -1.5279233 -1.7184368       -1.7184368                NA       -1.7184368
## 10 -3.6457397 -1.8968813       -1.8968813        -3.6457397       -1.8968813
##    flatterOutput.var
## 1                 NA
## 2          -2.105576
## 3          -1.708810
## 4          -1.071710
## 5                 NA
## 6                 NA
## 7                 NA
## 8           1.244812
## 9          -1.527923
## 10                NA
```

Plot to See Clustering Squared Deviations

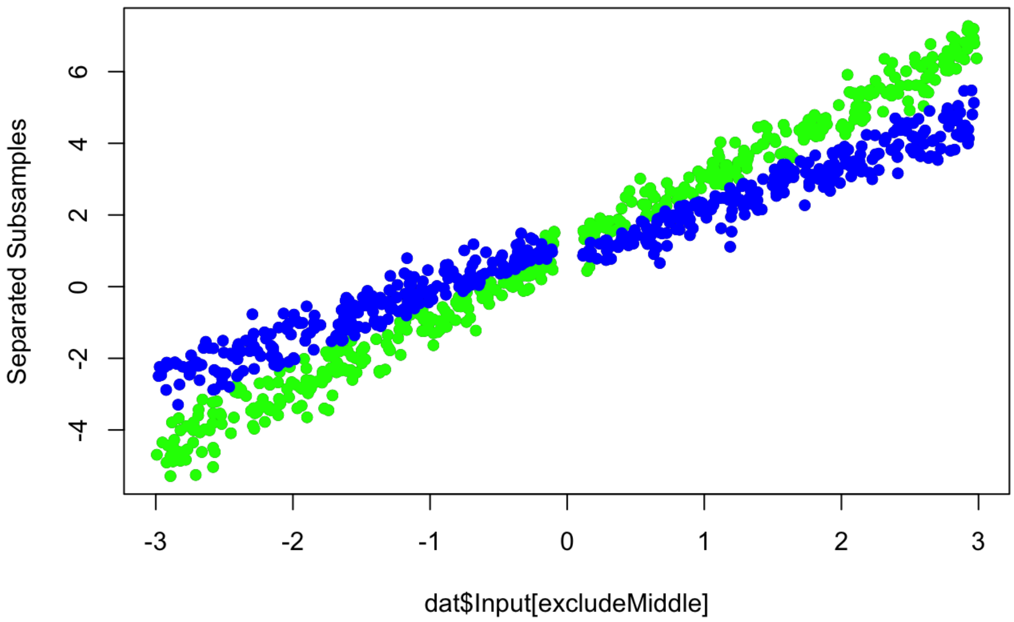
```
plot(dat$Input,
     (dat$Output-mean(dat$Output))^2,
     type="p",pch=19,ylab="Squared Deviations")
points(dat$Input,clusteringParabola,pch=19,col="red")
points(dat$Input[Unscrambling.Sequence.Steeper.var],
       (dat$Output[Unscrambling.Sequence.Steeper.var]-
          mean(dat$Output))^2,
       pch=19,col="blue")
points(dat$Input[!Unscrambling.Sequence.Steeper.var],
       (dat$Output[!Unscrambling.Sequence.Steeper.var]-
          mean(dat$Output))^2,
       pch=19,col="green")
```



#Plot to See Clustering

Regression for steeper and flatter

```
excludeMiddle<-(dat$Input<=mean(dat$Input)-0.1)|
  (dat$Input>=mean(dat$Input)+0.1)
matplot(dat$Input[excludeMiddle],cbind(dat$Output[excludeMiddle],
                                       Subsample.Steeper.var$steeperOutput.var[excludeMiddle],
                                       Subsample.Flatter.var$flatterOutput.var[excludeMiddle]),
        type="p",col=c("black","green","blue"),
        pch=16,ylab="Separated Subsamples")
```



Create lm object for the steeper and the flatter respectively. Check for results.

```
mSteep <- lm(Subsample.Steeper.var$steeperOutput.var~Subsample.Steeper.var$steeperInput.var,data = Subsample
.Steeper.var)
mFlat <- lm(Subsample.Flatter.var$flatterOutput.var~Subsample.Flatter.var$flatterInput.var,data = Subsample.
Flatter.var)
rbind(mSteep$coefficients,mFlat$coefficients)
```

```
##      (Intercept) Subsample.Steeper.var$steeperInput.var
## [1,]    1.013259                               1.930670
## [2,]    1.015640                               1.192568
```