Assignment 10

Air Quality data set

→

Handle the missing values:

Load data in to airqual variable.

```
> airqual <- read.csv(file.choose(), header = TRUE, sep = ",")
```

Replace the (-200) values with the 'NA' values in the dataset for all the fields.

```
> airqual$CO.GT.<- ifelse(airqual$CO.GT.==-200,NA,airqual$CO.GT.)
> airqual$PT08.S1.CO.<- ifelse(airqual$PT08.S1.CO.==-200,NA,airqual$PT08.S1.CO.)
```

Check the missing data using VIM library

```
> library(VIM)
> missingdata <- aggr(airqual)
> missingdata
```

```
 Missings in variables:
     Variable Count
     NMHC.GT.  8557
```

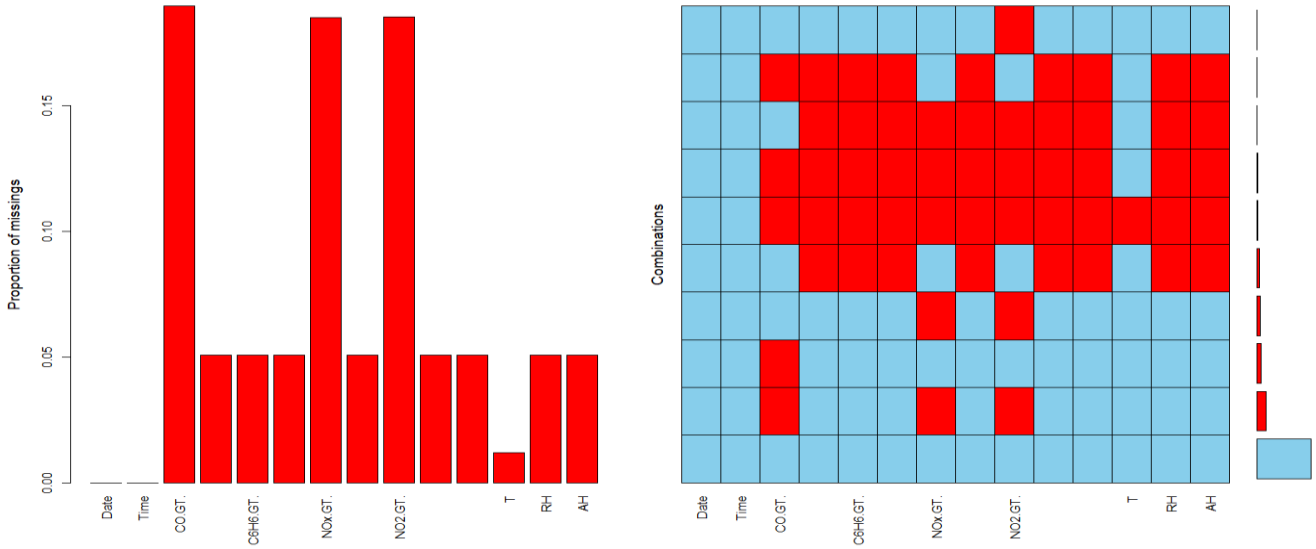
```
> # Approximately 90% of data from 'NMHC.GT.' culumn is missing thus remove the column

> airqual <- airqual[,-5]
```

Check missing values again

```
> missingdata <- aggr(airqual)
> missingdata
```

```
 Missings in variables:
     Variable Count
      CO.GT.  1797
   PT08.S1.CO.   480
     C6H6.GT.   480
 PT08.S2.NMHC.   480
      NOx.GT.  1753
  PT08.S3.NOx.   480
      NO2.GT.  1756
  PT08.S4.NO2.   480
   PT08.S5.O3.   480
          T   114
```

RH 480
AH 480



Impute the missing data using the MICE package using predictive mean matching method.

```
library(mice)

# Impute data using PMM method

airqual_imput <- mice(airqual,m=2,maxit=50,meth='pmm',seed=500)
```

Remove records with Date field as 'NA'

```
ImputedAirqual <- ImputedAirqual[complete.cases(ImputedAirqual), ]
```

Merging Date and Time column form the data set as below:

```
ImputedAirqual$Date_time <-
as.POSIXct(strptime(paste(ImputedAirqual$Date,ImputedAirqual$Time,sep=' '), "%m/%d/%Y
%H:%M:%S"))
```

Remove

```
ImputedAirqual <- ImputedAirqual[complete.cases(ImputedAirqual), ]
```

Create Time series data set as follows:

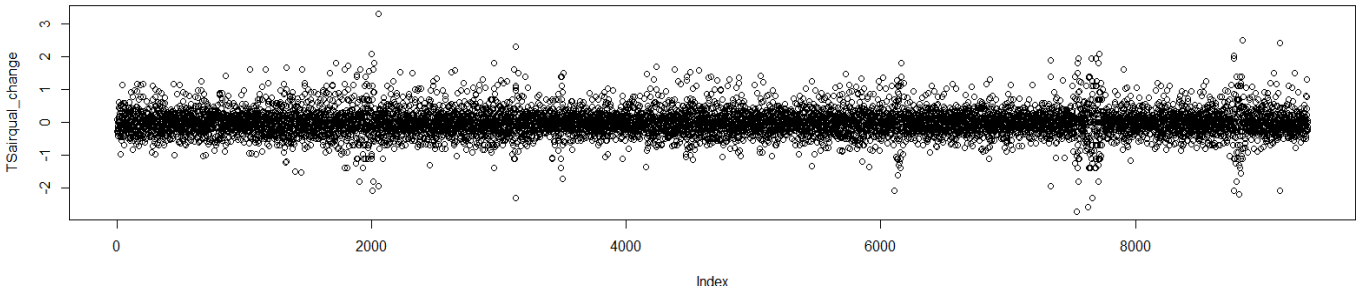
```
> DateTS <- as.POSIXlt(ImputedAirqual[,15], format = "%Y-%m-%d %H:%M:%S") #create date and time o
bjects
```

Create a time series data frame for air quality indicator "CO.GT.".

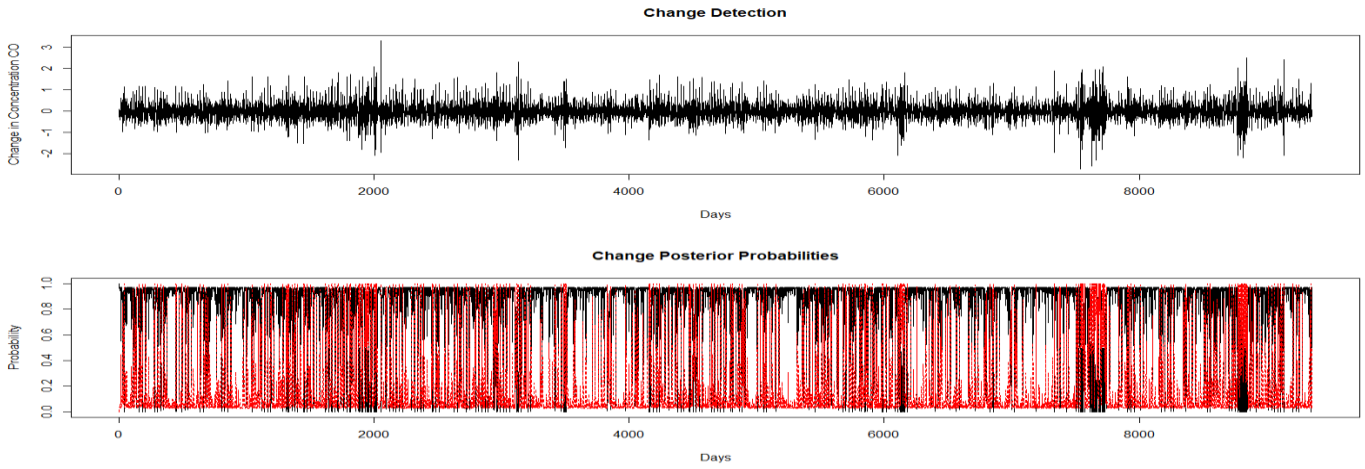
```
> TSairqual<-data.frame(ImputedAirqual[,3],row.names=DateTS)
> TSairqual<-as.xts(TSairqual) #build our time series data set
```

Create the air quality change matrix using the differential and logarithmic functions as below:

```
> colnames(TSairqual)<-c("CO.GT.Close")
> TSairqual_change = diff(log(Cl(TSairqual)))

> TSairqual_change = as.numeric(TSairqual_change)
> plot(TSairqual_change)
```

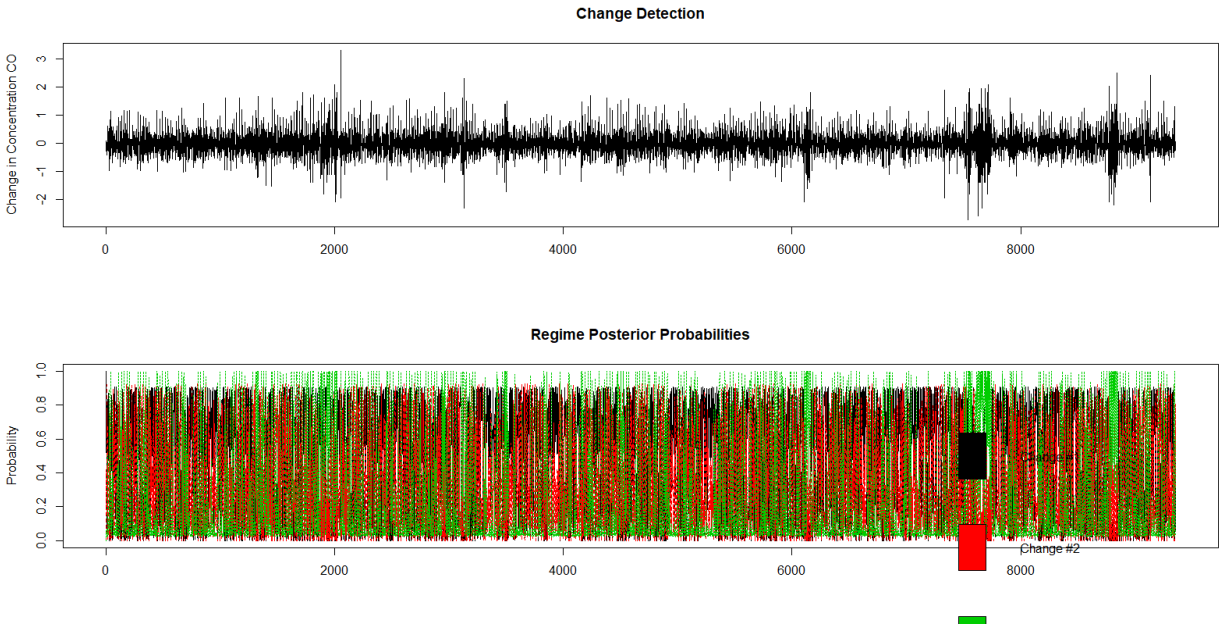



```
> # Fit a Hidden Markov Model with two states
> # to the S&P500 returns stream
> hmm <- depmix(TSairqual_change ~ 1, family = gaussian(), nstates = 2, data=data.frame(TSairqual_cha
nge=TSairqual_change))
> hmmfit <- fit(hmm, verbose = FALSE)
converged at iteration 56 with logLik: -3711.242
> post_probs <- posterior(hmmfit)
> # Plot the returns stream and the posterior
> layout(1:2)
> plot(TSairqual_change, type='l', main='Change Detection', xlab='Days', ylab='Change in Concentration
CO')
> matplot(post_probs[,-1], type='l', main='Change Posterior Probabilities', xlab='Days', ylab='Probability')
```



```
# Fit a Hidden Markov Model with three states

# to the S&P500 returns stream

hmm <- depmix(TSairqual_change ~ 1, family = gaussian(), nstates = 3,
data=data.frame(returns=returns))

hmmfit <- fit(hmm, verbose = FALSE)

post_probs <- posterior(hmmfit)
```

```
> # Fit a Hidden Markov Model with three states
> # to the S&P500 returns stream
> hmm <- depmix(TSairqual_change ~ 1, family = gaussian(), nstates = 3, data=data.frame(TSairqual_cha
nge=TSairqual_change))
> hmmfit <- fit(hmm, verbose = FALSE)

> layout(1:2)
> plot(returns, type='l', main='Change Detection', xlab='', ylab='Change in Concentration CO')
> matplot(post_probs[,-1], type='l', main='Regime Posterior Probabilities', ylab='Probability')
> legend(x='topright', c('Change #1','Change #2', 'Change #3'), fill=1:3, bty='n')
```



Since the data is hourly, we can see significant change in the CO concentration approximately at 2000 hours (2000/24 ~ 84) i.e. around 3rd month in the data which is around Aug 2004. Also, there is large and continuous change at the 7500 hours to 8000 hours i.e. 10th month in the data which is around Jan 2005.

To get the clear picture we can convert the data into daily level by measuring the Opening, closing, maximum and minimum concentration of CO for the day. We can aggregate data to find minimum and maximum value of CO for the day. Also, Open value can be considered as the value at time 0:00:00 and close value could be 23:00:00.

Thus, the data set will look like this:

```
> names(AirQualTMS)
[1] "Date"        "CO.GT.Open"  "CO.GT.Max"   "CO.GT.Min"   "CO.GT.Close"
```

Aggregation code:

```
> aggdata_min <-aggregate(ImputedAirqual$CO.GT., by=list(ImputedAirqual$Date)
,
+                       FUN=min, na.rm=TRUE)
> aggdata_max <-aggregate(ImputedAirqual$CO.GT., by=list(ImputedAirqual$Date)
,
+                           FUN=max, na.rm=TRUE)
> airqual_open <- ImputedAirqual[ImputedAirqual$Time=='0:00:00',]
> airqual_close <- ImputedAirqual[ImputedAirqual$Time=='23:00:00',]
```

Merge the data into one data frame

```
> Merged_data <- merge(aggdata_max, aggdata_min, by="Date")
> Merged_data <- merge(Merged_data, airqual_close, by="Date")
> Merged_data <- merge(Merged_data, airqual_open, by="Date")
```

Create time series dataset for aggregated day level data:

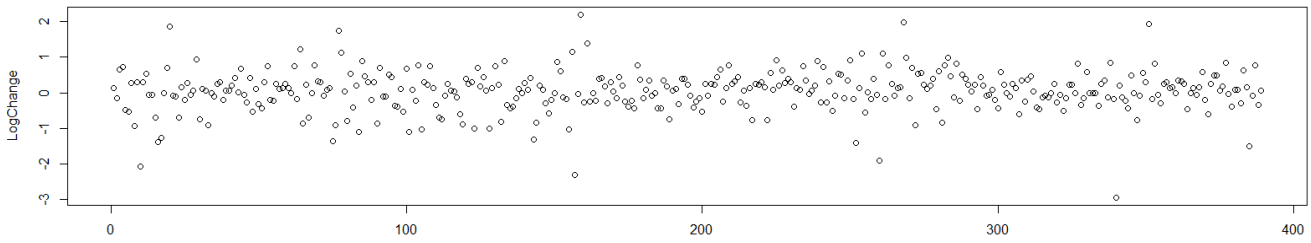
```
> Date<-as.character(AirQualTMS_cleans[,1])
> DateTS<- as.POSIXlt(Date, format = "%m/%d/%Y") #create date and time object
s
> TSData<-data.frame(AirQualTMS_cleans[,2:5],row.names=DateTS)
> TSData<-as.xts(TSData) #build our time series data set
```

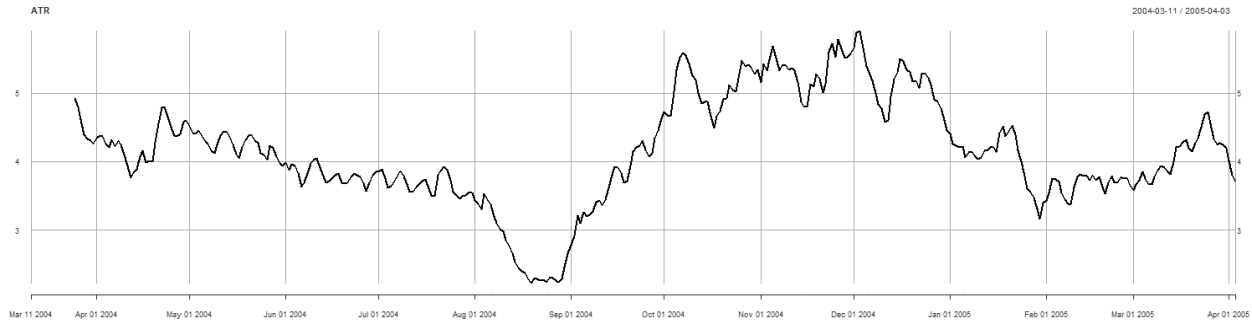
Calculate Average True Range (ATR) and log change for Open and close values as below:

```
> ATRindicator<-ATR(TSData[,2:4],n=14) #calculate the indicator
> ATR<-ATRindicator[,2] #grab just the ATR
> LogChange <- log(AirQualTMS_cleans$CO.GT.Close) - log(AirQualTMS_cleans$CO.
GT.Open)


> ModelData<-data.frame(LogChange,ATR) #create the data frame for our HMM mod
el
> colnames(ModelData)<-c("LogChange","ATR") #name our columns
```

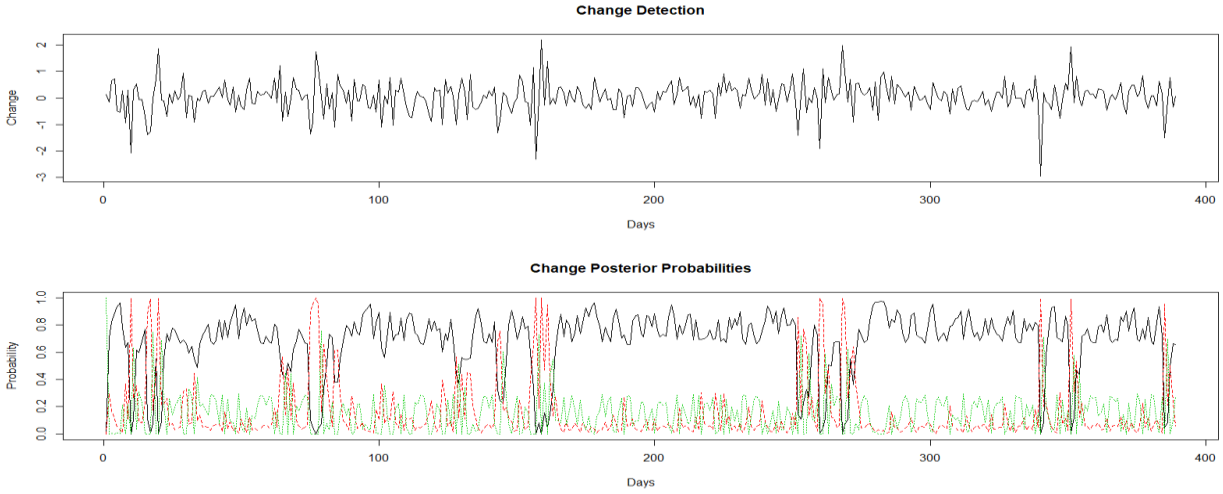
Plot the ATR and LogChange:

```
> plot(LogChange)
> plot(ATR)
```

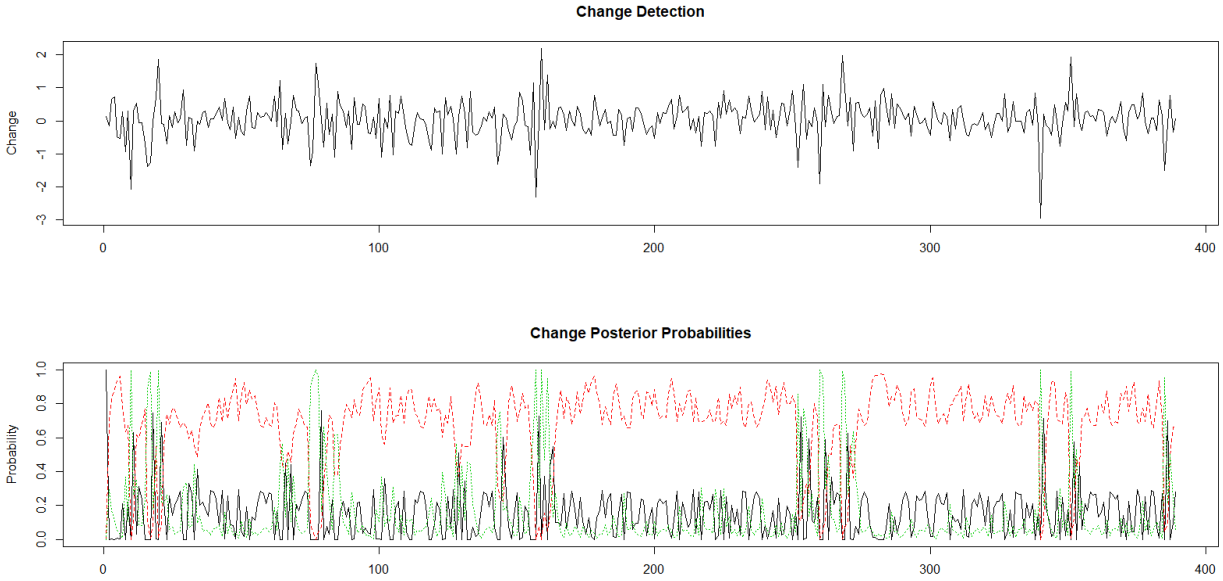





```
> hmm <- depmix(ATR~1, family = gaussian(), nstates = 3, data=ModelData)
> hmmfit <- fit(hmm, verbose = FALSE)
converged at iteration 26 with logLik: -150.2579
> post_probs <- posterior(hmmfit)
> layout(1:2)
> plot(ModelData$LogChange, type='l', main='Regime Detection', xlab='Days', y
lab='Returns')
> matplot(post_probs[,-1], type='l', main='Regime Posterior Probabilities', x
lab='Days', ylab='Probability')
```



```
> # Fit a Hidden Markov Model with three states
> # to the S&P500 returns stream
> hmm <- depmix(LogChange ~ 1, family = gaussian(), nstates = 3, data=ModelDa
ta)
> hmmfit <- fit(hmm, verbose = FALSE)
converged at iteration 490 with logLik: -297.7593
> post_probs <- posterior(hmmfit)
> layout(1:2)
> plot(ModelData$LogChange, type='l', main='Change Detection', xlab='', ylab=
'Change')
> matplot(post_probs[,-1], type='l', main='Change Posterior Probabilities', y
lab='Probability')
```



From the daily HMM model, we can see that there significant variation around 90th day, 150th day and 340th day.