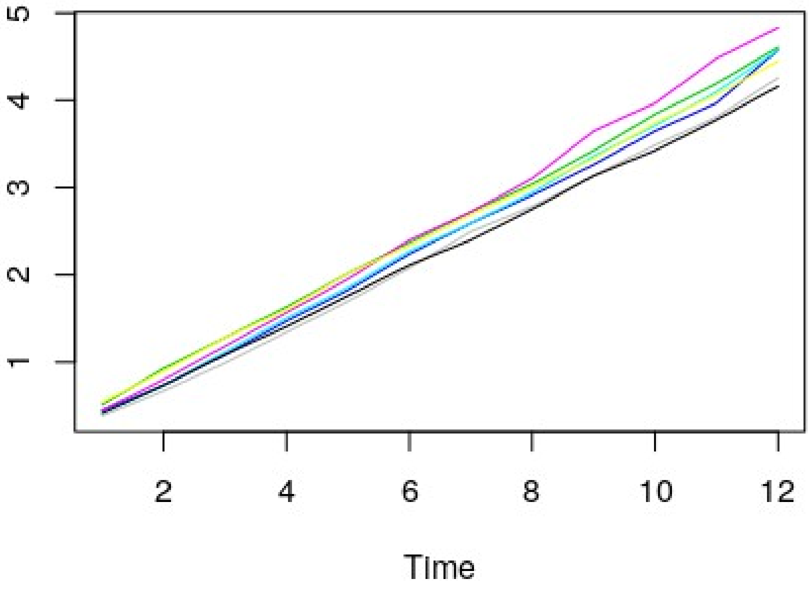
hw2

1 (a)

line plot for each measure labelled V and X is the number of observation

```
music_data <-
read.csv('/home/shreyas/Documents/Masters_stuff/IU_assignments/Da
ta_mining/rachmaninov_pc2_onset.csv')
#music_data <- t(music_data)
music_data <- music_data[,2:13]
ts.plot(t(music_data), col=3:15)
```



1(b)

Covariance using matrix multiplication is equal to the orginal covariance of X

```
music_data <-
read.csv('/home/shreyas/Documents/Masters_stuff/IU_assignments/Da
ta_mining/rachmaninov_pc2_onset.csv')
```

```
music_data <- music_data[,2:13]

cov_music <- cov(music_data)

music_means <- colMeans(music_data)

music_data_minus_means <- t(t(music_data) - music_means)

Calculated_cov <- (t(music_data_minus_means) %*%
music_data_minus_means)/6

inds <- all.equal(Calculated_cov, cov_music)
inds

## [1] TRUE
```

1(c)

Created a matrix of n = 1000 and columns equal to number of columns of X and using rnorm function of mean 0 and sd as 1. Then multiplying each column with their directional variance of cov(X)

```
music_data <-
read.csv('/home/shreyas/Documents/Masters_stuff/IU_assignments/Da
ta_mining/rachmaninov_pc2_onset.csv')
music_data <- music_data[2:13]
music_means <- colMeans(music_data)
cov_music <- cov(music_data)
svd_music <- svd(cov_music, nu=12)

thousand_points <- rnorm(1000)

for(i in 1:12){

  colm <- rnorm(1000, sd=sqrt(var(music_data[i])))
  thousand_points <- cbind(thousand_points, colm)
}

thousand_points <- thousand_points[,2:13]
new_points <- thousand_points %*% svd_music$u

new_points <- t(t(new_points) + music_means)
```

1(d)

a) here the data has more standard deviation when taken on only one vector b)here the data has less standard deviation and will keep on reducing as we add more and more axis to project data on.

3

Scaled the matrix and take svd of the cov. Calculated the most variance covered by d and created a new vector by matrix multiplication of orginal matrix with U and selected first 5 columns of it.

```
mystery <-
read.csv('~/Documents/Masters_stuff/IU_assignments/Data_mining/my
stery.csv')
mystery_means <- colMeans(mystery)
k <- t(mystery) - mystery_means
mystery_minus <- t(k)

cov_mystery <- cov(mystery_minus)
svd_cov <- svd(cov_mystery)
#here the first 5 di have a value greater than 0, so lets
consider the first five vectors
svd_cov_U <- svd_cov$u
newU <- svd_cov_U[,1:5]
reduced_cord <- mystery_minus %*% newU
```

4

-Divided the dataset into good class and bad class -Calculated the mahalanobis distance between the means and all the points - Compared the distances and made a true false table -added the true values to calculate the true positives and the true negatives -and calculated the flase positives and false negatives

```
library('mlbench')
data('Ionosphere')
data <- Ionosphere
data$V1 <- as.numeric(as.character(data$V1))
data$V2 <- as.numeric(as.character(data$V2))
classes <- data['Class']

data_good <- data[data['Class'] == 'good',]
data_bad <- data[data['Class'] == 'bad',]

#-----------------------------------------------------------------
-------------
#calculate the mahalanobis distance of each class from the mean
of each class
```

```
#then assign to classes according to distance from each of the
means
#----------------------------------------------------------------
-------------
# to be changed from here

data_sliced_good <- data_good[,3:34]
good_means <- colMeans(data_sliced_good)

data_sliced_bad <- data_bad[,3:34]
bad_means <- colMeans(data_sliced_bad)

#Good data set
#Good with bad mean
mahala_g_b_mean <- mahalanobis(data_sliced_good, bad_means,
cov(data_sliced_good))
#Good with good mean
mahala_g_g_mean <- mahalanobis(data_sliced_good, good_means,
cov(data_sliced_good))

#Bad data set
#bad with bad mean
mahala_b_b_mean <- mahalanobis(data_sliced_bad, bad_means,
cov(data_sliced_bad))
#bad with good mean
mahala_b_g_mean <- mahalanobis(data_sliced_bad, good_means,
cov(data_sliced_bad))

bad_pred_ind <- mahala_b_b_mean < mahala_b_g_mean
good_pred_ind <- mahala_g_g_mean < mahala_g_b_mean

data_good_new <- cbind(data_good, good_pred_ind)
data_bad_new <- cbind(data_bad, bad_pred_ind)


G_G_ind <- sum(bad_pred_ind)    #True positives
B_B_ind <- sum(good_pred_ind)   #True Negatives
G_B_ind <- 126 - G_G_ind        #False Negatives
B_G_ind <- 225 - B_B_ind        #False Positives
```