1 National Track Records for Men and Women

(a) Exploratory Data Analysis of Women's Records

The data contains fastest known times for different length races for 54 different countries. There are records for the 100m, 200m, 400m, 800m, 1500m, 3000m, and the Marathon for each country. Refer to summary statistics in this section for general information. While assessing the univariate normality, figure 2 suggests the densities appear to become more and more right skewed as the distance of the run increases. Figure 2 (histograms), Figure 3 (Q-Q plot), and Figure 4 (Q-Q plot) suggests that marginal univariate normality appears to be violated for the 1500m, 3000m, Marathon, log1500m, log3000m, and logMarathon. However, normality does not seem to be violated for the 100m, 200m, 400m, and 800m.

The pairwise bivariate contour plot (Figure 6) and the Chi-square plot (Figure 5) for the 100m, 200m, 400m, 800m densities do not provide any significant evidence against multivariate normality. However, multivariate normality seems to be violated when including any of the 1500m, 3000m, or Marathon densities.

The correlation matrix and scatterplot matrix suggest that the correlation is proportional to the difference in distance between the run. As the distance between the runs decreases, the correlation between them increases. The scatterplot matrix suggests that there is a stronger linear relationship between runs that have smaller differences in distance.

In summary, the 100m, 200m, 400m, 800m appear to be multivariate normal. The marginal densities of the 1500m, 3000m, and Marathon do not appear to be normal. Interestingly, as the distance of the run type increases the density becomes increasingly right-skewed.

(b) Probability Ellipse and Confidence Regions

Figure 7 contains the .50 and .95 probability ellipses. There are five instances that do not lie in the .95 probability ellipse. There are about 19 instances that do not lie in the .50 probability Ellipse.

Figure 8 displays the 95% confidence ellipse for the mean vector, the 95% simultaneous T^2 confidence region, the 95% simultaneous Bonferroni confidence region, and the individual 95% confidence intervals for the mean of one variable. The confidence ellipse is used if we want to consider the confidence "space" of the mean vector. The Hotelling confidence region is most conservative and is used if you want to consider both means together while considering the effects of all variables. The Bonferroni confidence region is used if we want to simply consider one mean while considering the effects of all variables. The individual confidence interval is used if we want to consider one variables mean without the effects of the other variables. It is the least conservative.

(c) Mean Comparison between Male and Female

We are going to simply compare the individual means between the two populations: Men and Female. Typical assumptions include (1) the density of each variable for each population (sex) is normally distributed, (2) the samples between sexes were independently drawn, and (3) the populations share the same variance-covariance. Luckily, we have a large number of samples. The analysis of variance model is

robust to non-normal data with a large number of samples. We will assume the samples between sexes were independently sampled. Furthermore, if we simply use the adhoc method to compare variance, Table 2 and Table 3 suggest each of the individual variances between each population for a particular distance is less than a factor of 4. Thus, assumptions are met and we can perform analysis of variance. The results lie in Tables 5-9 and suggest there is significant evidence to claim the mean times for each distance between sexes are significantly different.

(d) Principal Components Analysis for Men

We perform principal components analysis on the correlation matrix since the units are different for each variable i.e. the Marathon is measured in minutes and the 100m in seconds.

The first two principal components capture 94% of the variance. These are sufficient in explaining the behavior of the distribution. If we standardize the data and multiply by the first eigenvector and the second eigenvector, then we will get the corresponding first and second principal components. A biplot if provided in Figure 9 to illustrate the bivariate distribution.

Furthermore, if we rank the first principal components in descending order we will have the fastest countries. It is important to note that the first eigenvector which 'constructs' the first principal component has all negative elements. This means that the Country with fastest times (smallest times) will have the largest first principal component. So, the fastest countries corresponds to the countries with highest first principal components. Below illustrates the top ten. Intuitively, these are the countries I would expect to be on the top 10.

	Country	PC1
1	Great_Britain	14.01
2	USA	11.89
3	Italy	10.35
4	Germany	10.31
5	France	9.91
6	Kenya	9.37
7	Australia	8.90
8	Belgium	8.79
9	Russia	8.44
10	New_Zealand	8.01

Table 1: Women: Summary Statistics

	X100m	X200m	X400m	X800m
1	Min. :10.49	Min. :21.34	Min. :47.60	Min. :1.890
2	1st Qu.:11.12	1st Qu.:22.57	1st Qu.:49.97	1st Qu.:1.970
3	Median :11.32	Median :22.98	Median :51.65	Median :2.005
4	Mean :11.36	Mean :23.12	Mean :51.99	Mean :2.022
5	3rd Qu.:11.57	3rd Qu.:23.61	3rd Qu.:53.12	3rd Qu.:2.070
6	Max. :12.52	Max. :25.91	Max. :61.65	Max. :2.290

	X1500m	X3000m	Marathon
1	Min. :3.840	Min. : 8.100	Min. :135.2
2	1st Qu.:4.003	1st Qu.: 8.543	1st Qu.:143.5
3	Median :4.100	Median : 8.845	Median :148.4
4	Mean :4.189	Mean : 9.081	Mean :153.6
5	3rd Qu.:4.338	3rd Qu.: 9.325	3rd Qu.:157.7
6	Max. :5.420	Max. :13.120	Max. :221.1

Table 2: Men: Covariance

	X100m	X200m	X400m	X800m	X1500m	Marathon
X100m	0.12	0.21	0.43	0.02	0.04	1.69
X200m	0.21	0.42	0.79	0.03	0.08	3.56
X400m	0.43	0.79	2.09	0.08	0.19	9.49
X800m	0.02	0.03	0.08	0.00	0.01	0.48
X1500m	0.04	0.08	0.19	0.01	0.02	1.25
Marathon	1.69	3.56	9.49	0.48	1.25	86.37

Table 3: Women: Covariance

	X100m	X200m	X400m	X800m	X1500m	Marathon
X100m	0.16	0.34	0.89	0.03	0.08	4.33
X200m	0.34	0.86	2.19	0.07	0.20	10.38
X400m	0.89	2.19	6.75	0.18	0.51	28.90
X800m	0.03	0.07	0.18	0.01	0.02	1.22
X1500m	0.08	0.20	0.51	0.02	0.07	3.54
Marathon	4.33	10.38	28.90	1.22	3.54	270.27

Table 4: Women: Correlation

	X100m	X200m	X400m	X800m	X1500m	X3000m	Marathon
X100m	1.00	0.94	0.87	0.81	0.78	0.73	0.67
X200m	0.94	1.00	0.91	0.82	0.80	0.73	0.68
X400m	0.87	0.91	1.00	0.81	0.72	0.67	0.68
X800m	0.81	0.82	0.81	1.00	0.91	0.87	0.85
X1500m	0.78	0.80	0.72	0.91	1.00	0.97	0.79
X3000m	0.73	0.73	0.67	0.87	0.97	1.00	0.80
Marathon	0.67	0.68	0.68	0.85	0.79	0.80	1.00

Table 5: AOV: 100m

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender	1	20.95	20.95	150.04	0.0000
Residuals	106	14.80	0.14		

Table 6: AOV: 200m

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender	1	126.86	126.86	198.18	0.0000
Residuals	106	67.85	0.64		

Table 7: AOV: 400m

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender	1	821.05	821.05	185.84	0.0000
Residuals	106	468.32	4.42		

Table 8: AOV: 800m

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender	1	1.40	1.40	241.87	0.0000
Residuals	106	0.61	0.01		

Table 9: AOV: 1500m

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender	1	6.43	6.43	130.76	0.0000
Residuals	106	5.22	0.05		

Table 10: AOV: Marathon

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gender	1	7724.15	7724.15	43.32	0.0000
Residuals	106	18902.11	178.32		

MS Patients and Visual Stimuli Groups

The goal is to develop a rule for separating people with MS from people who are not using a classification model. The data includes Age, Stimuli 1 (sum and difference), Stimuli 2 (sum and difference), and the MS indicator.

(a) Assess Marginal Normality

Some of the marginal densities did not appear to be normal. In particular, Age for MS=0, Stimuli 1 difference, Stimuli 2 difference did not appear to be normal. A power transformation on age and log transformation on non-zero Stimuli 1 difference and Stimuli 2 difference decently transform these variables to normality. Figure 10 and Figure 11 provide the post-transform Q-Q plots illustrating normality. One significant issue that has significantly delayed this report is that the Normality should be assessed across MS positive and negative i.e. if I transform Age in MS positive and not in MS negative and then build a model, then there will be no way of know how to transform new data.

(b) Fisher's Linear Discriminant

Assuming equal covariance between matrices, a Fisher's linear discriminant function is created. Using all of the variables in the model creates higher apparent error rate than other subsets. I simply looked at the apparent error rate for many different combinations of variables and found that the optimal subset of variables to consider is Stimuli 1 sum and Stimuli 2 sum. These seemed to created the best discriminant function. The rule

$$\hat{y} = \begin{bmatrix} -0.0158 & -.0756 \end{bmatrix} \begin{bmatrix} \text{S1sum} \\ \text{S2sum} \end{bmatrix} \leq -18.9170 = m \tag{1}$$

determines if x_0 should be classified at MS positive and will be MS negative otherwise. The associated apparent error rate is 12.24%. The ideal is to look at all combinations of variables considering the Expected Actual Error Rate.

(c) Lachenbruch's "Holdout Procedure"

This procedure estimates the expected actual error rate and provides a measure of generalization. We (1) loop through holding out every instances once, (2) build classification model, (3) classify holdout, (4) repeat for each instance and (5) divide the sum of all misclassified holdouts by total number of instances in data. Using this procedure on the model built with only the two variables– Stimuli 1 sum and Stimuli 2 sum– we get an expected actual error rate of .0918.

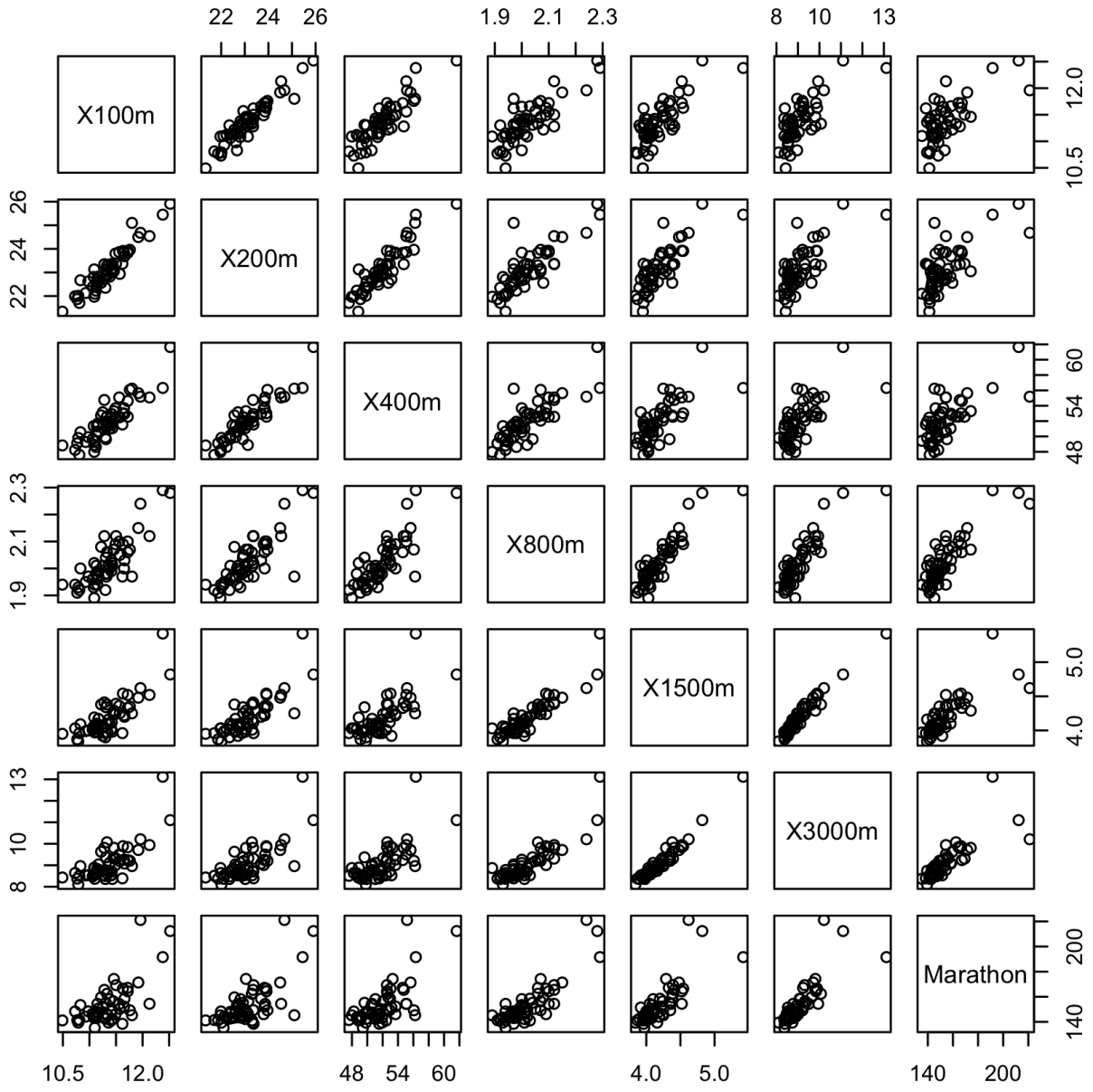

Figure 1: Scatter Plot Matrix: Each point represents a country. Each graph illustrates trends between pairs of races. Notice: generally, the closer the race distances are to each other the more correlated they appear to be.

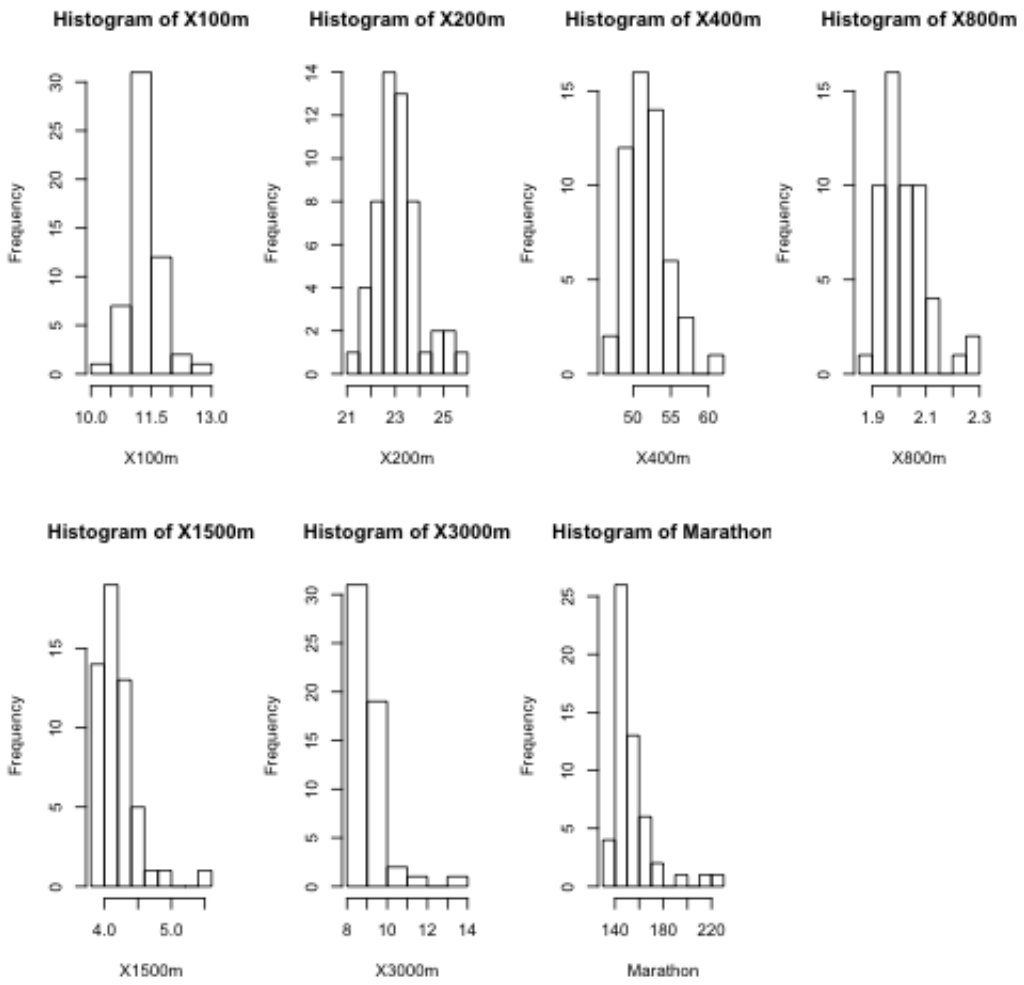

Figure 2: Histogram for each distance. The 100m, 200m, 400m, 800m appear normal whereas the 1500m, 3000m, and Marathon are right-skewed. The log-histogram plots of 1500m, 3000m, and the Marathon are also right=skewed.

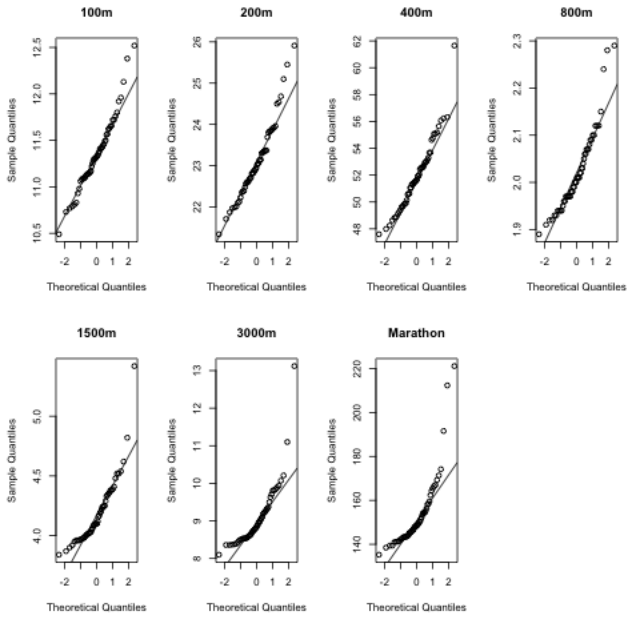

Figure 3: Q-Q plot for each distance. The densities for the 1500m, 3000m, and Marathon appear to be non-normal.

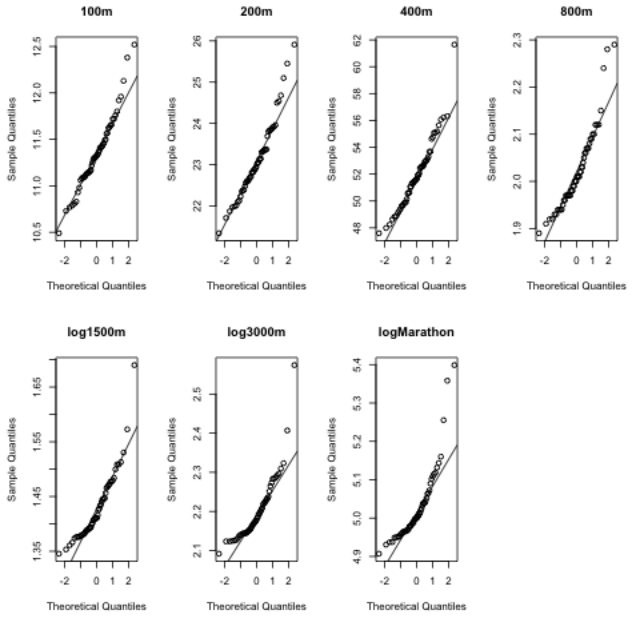

Figure 4: Q-Q plot for each distance. The densities for the log1500m, log3000m, and logMarathon even appear to be non-normal.

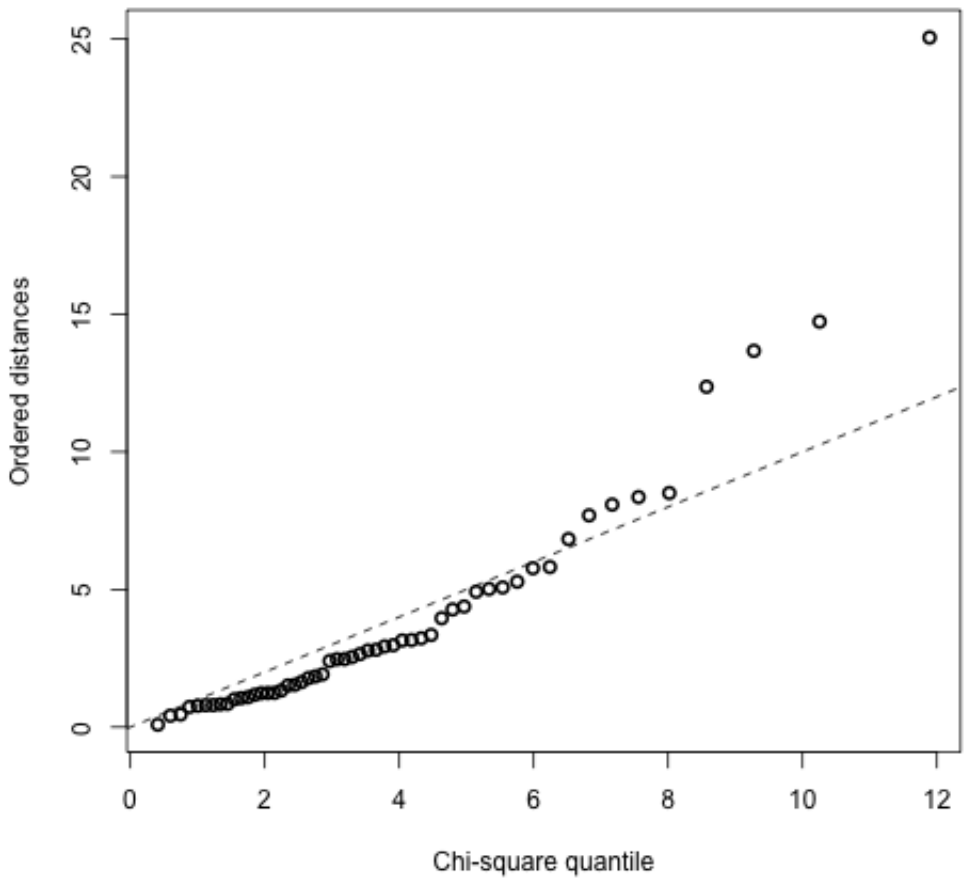

Figure 5: Chi-square plot for 100m, 200m, 400m, and 800m. The multivariate density does not appear to be non-normal

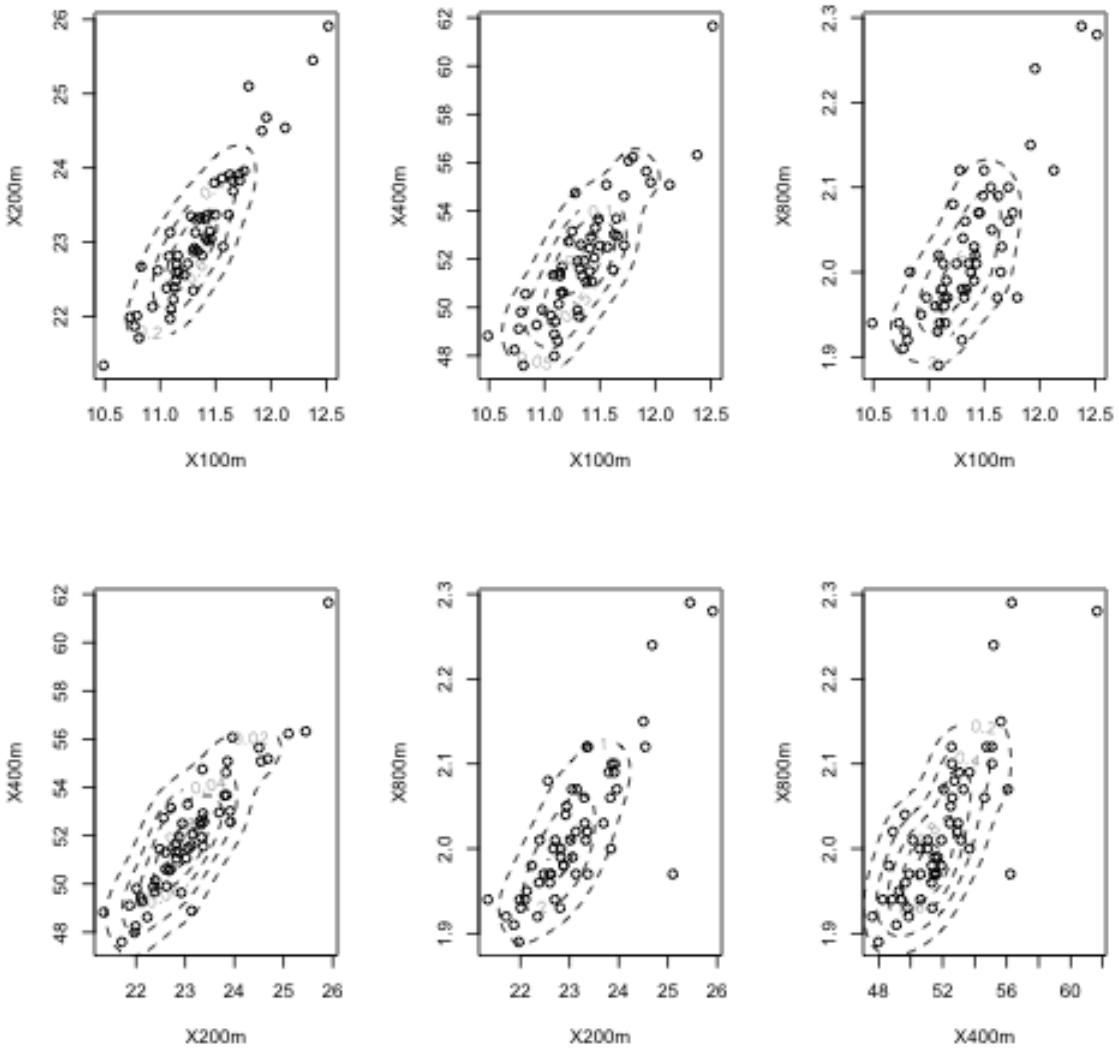

Figure 6: Contours of all six pairs of bivariate densities of 100m, 200m, 400m, and the 800m. No contour plots seems to suggest non-normality. The only potential concerns are the densities of the distances that are furthest apart.

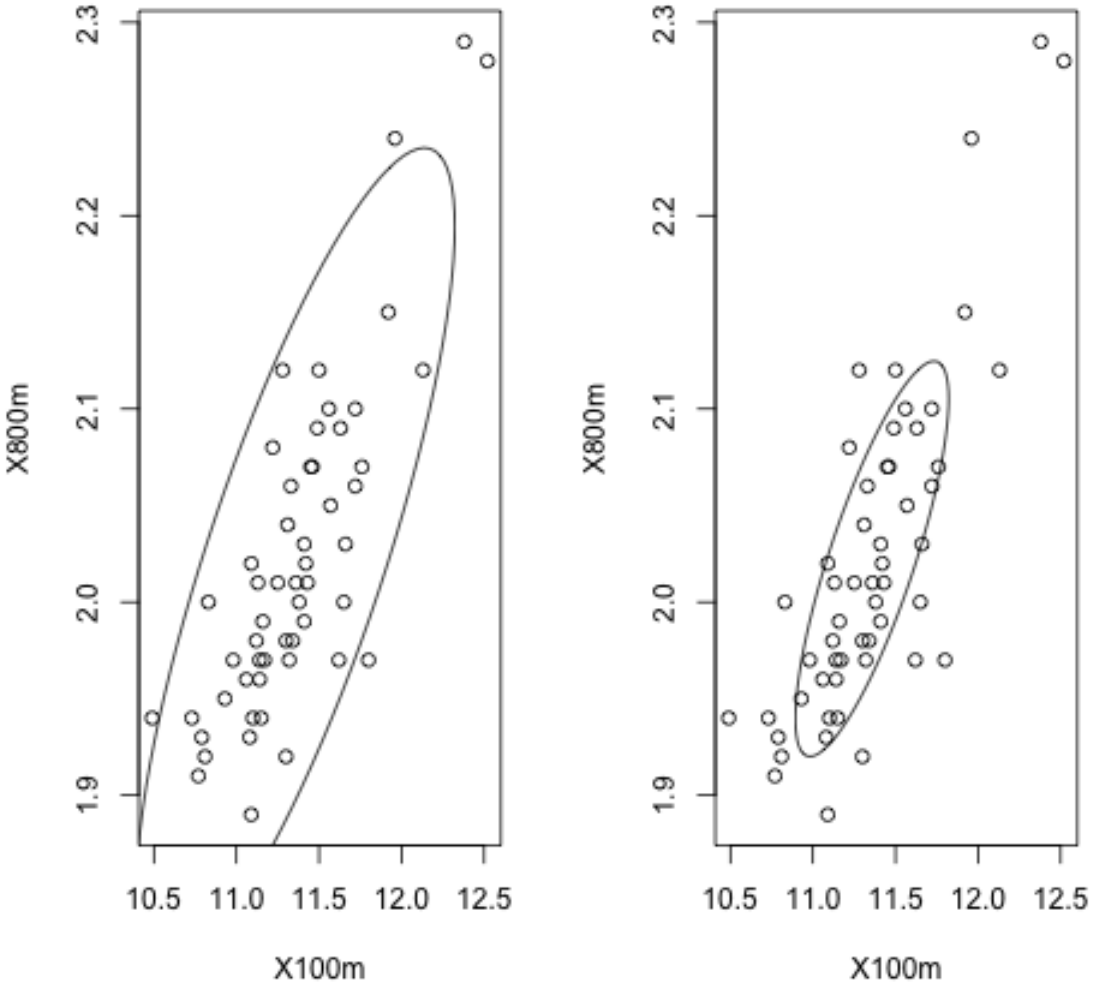

Figure 7: The left plot displays the .95 probability ellipse. The right plot displays the .50 probability ellipse.

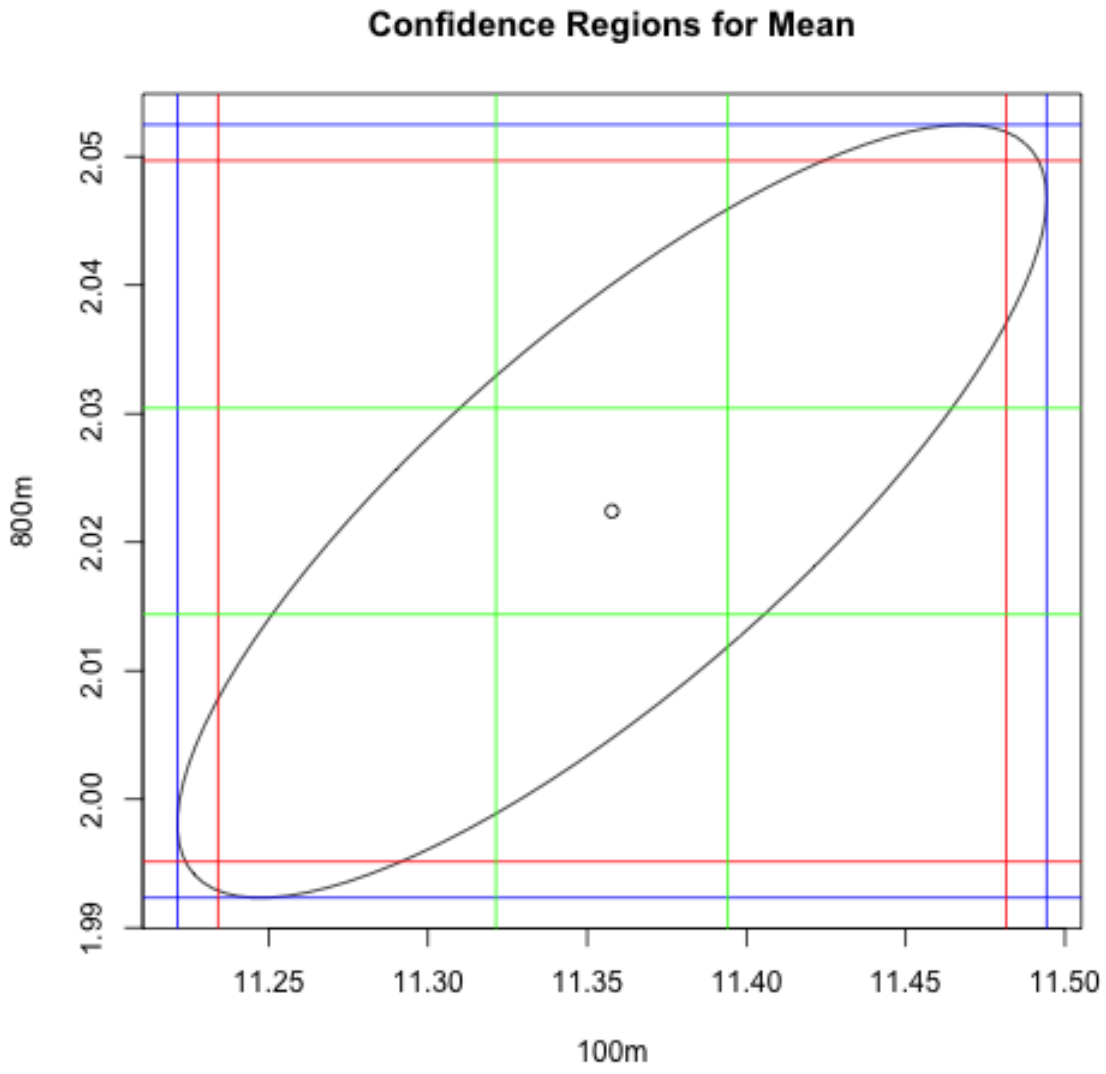

Figure 8: This figure displays various 95% confidence regions for the mean of 100m and 800m data. The outer rectangular region (blue) is the simultaneous Hotelling T^2 Interval, the middle (red) confidence rectangular regions are the simultaneous Bonferroni intervals for each mean, the inner (green) are the individual confidence intervals for each mean. The ellipse is the 95% confidence ellipse for the mean vector.

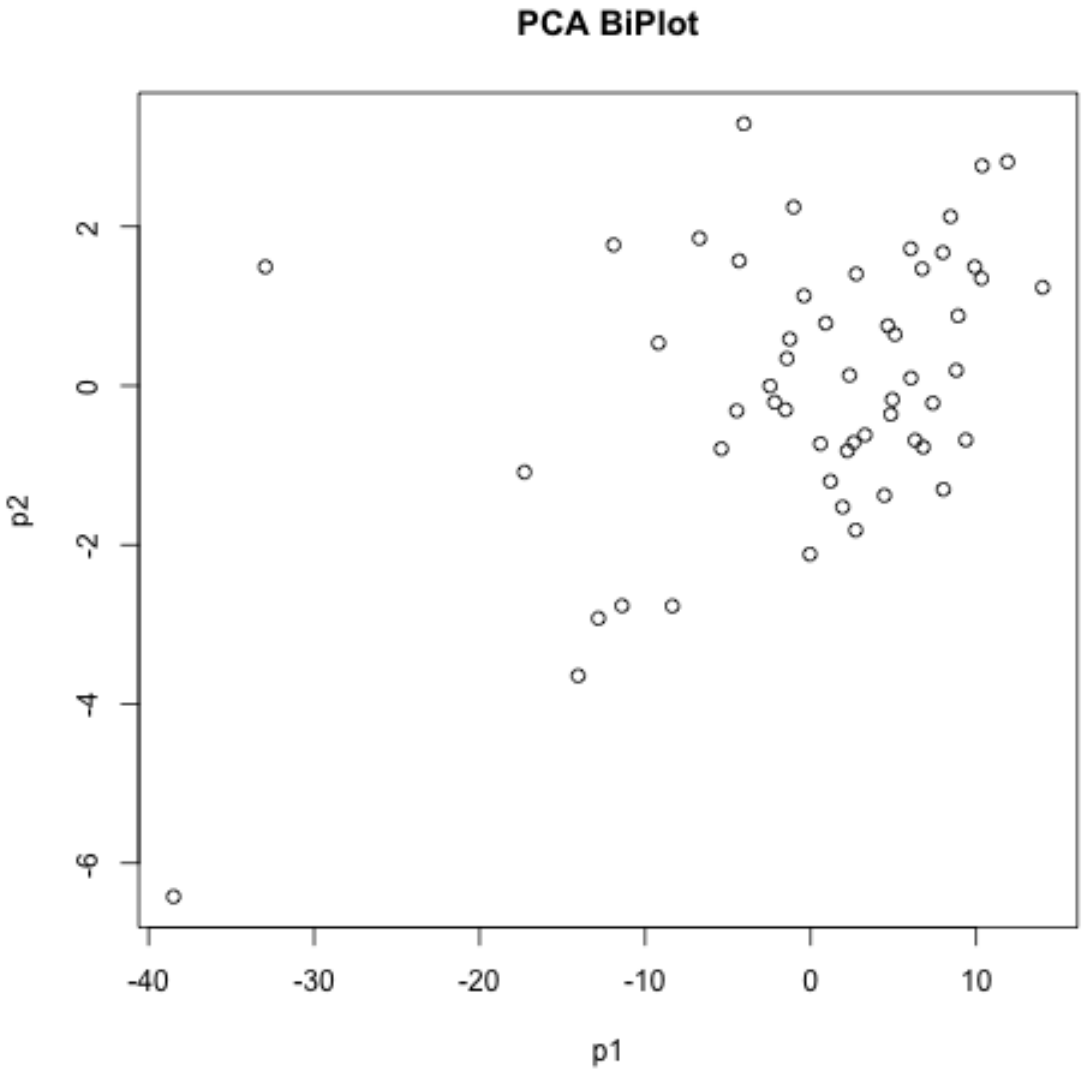

Figure 9: Simple biplot of the first two principal components

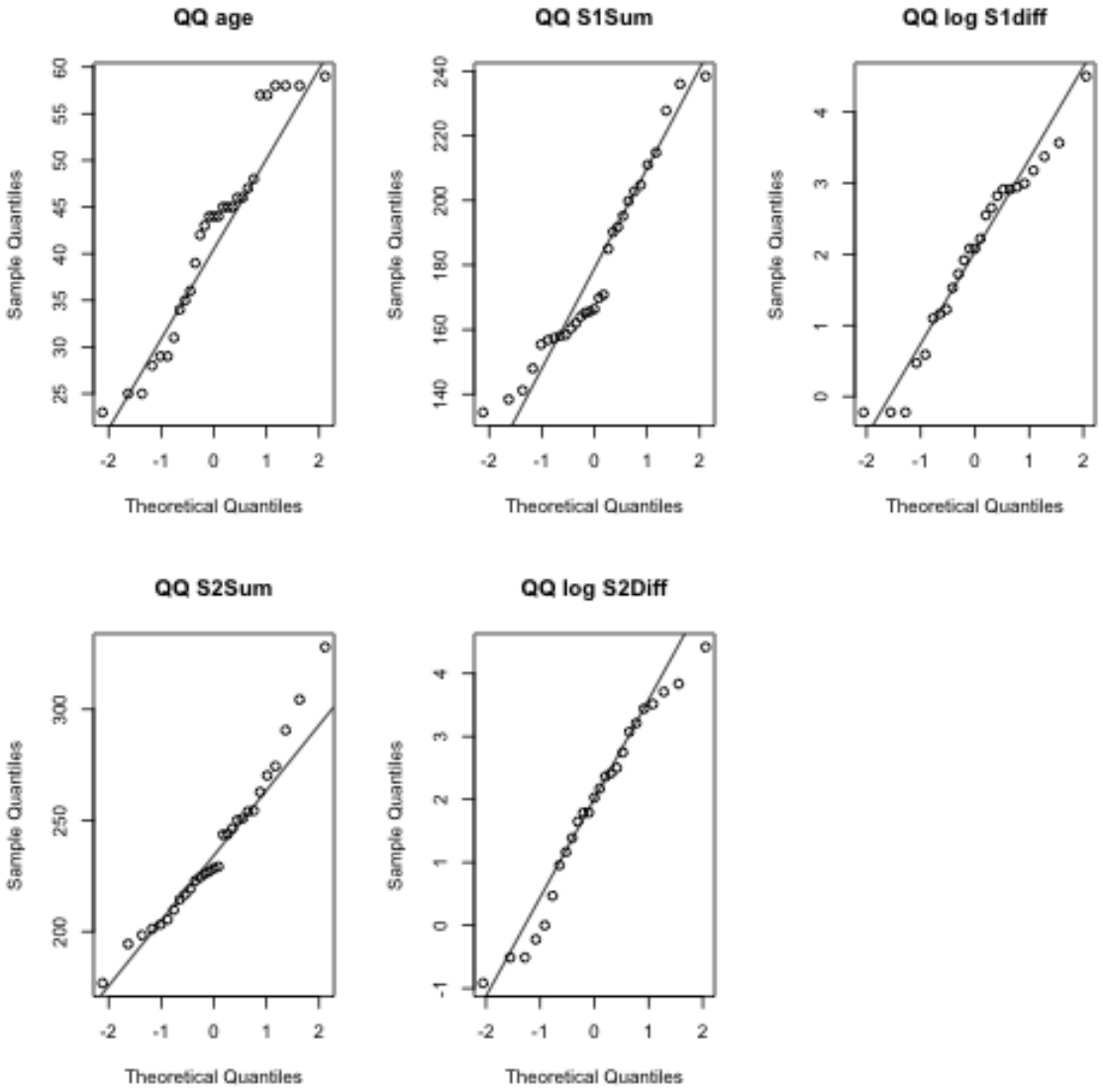

Figure 10: Q-Q plots of suggested transformed data of MS positive group

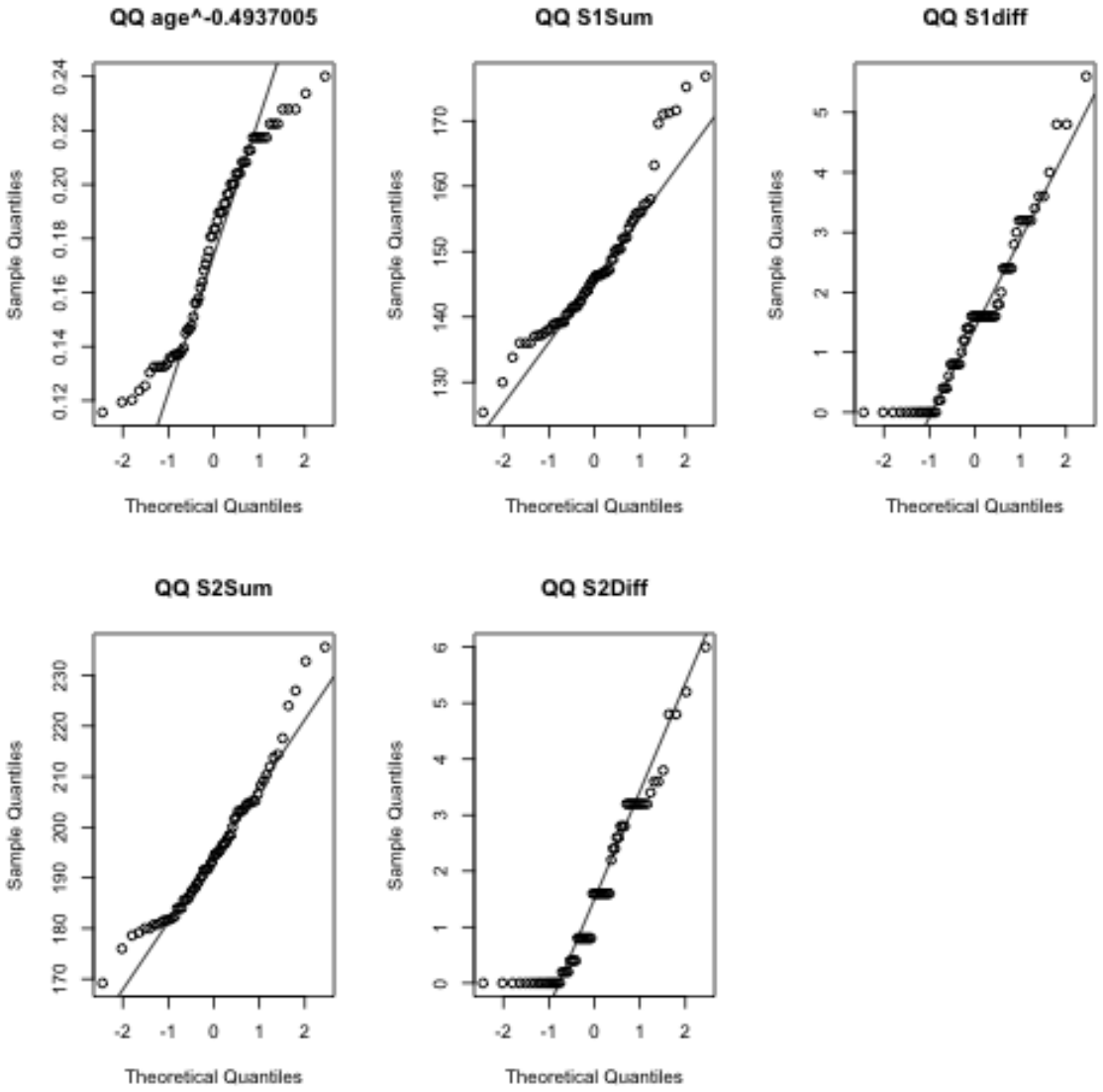

Figure 11: Q-Q plots of suggested transformed data of MS negative group

Code

1 (a)

```
################################
# Problem 1:  Part a code
################################

# used to generate tables for latex format
# install.packages("xtable")
library(xtable)

# save all output to a txt file.
sink("parta.txt", append=FALSE, split=FALSE)

# Load Data
Data=read.table('Women.txt', header=T);
X=Data[3:9];
attach(X);

# Summary Statistics: Two parts to fit on page
xtable(summary(X[1:4]))
xtable(summary(X[5:7]))

# Correlation Matrix
xtable(cor(X))

# generate png for matrix scatter plot iamge
png("a.scatterMatrix.png", width = 6, height = 6, units = 'in', res=300)
pairs(X)
dev.off()

#############################################################################
# Univariate  Normality Assessment

# Histograms
png("a.histograms.png")
par(mfrow = c(2,4))
hist(X100m)
hist(X200m)
hist(X400m)
hist(X800m)
hist(X1500m)
hist(X3000m)
hist(Marathon)
dev.off()
```

```
# Q-Q plots of each race
png("a.QQplots.png")
par(mfrow =c(2,4))
qqnorm(X100m, main ="100m")
qqline(X100m)
qqnorm(X200m, main ="200m")
qqline(X200m)
qqnorm(X400m, main ="400m")
qqline(X400m)
qqnorm(X800m, main ="800m")
qqline(X800m)
qqnorm(X1500m, main ="1500m")
qqline(X1500m)
qqnorm(X3000m, main ="3000m")
qqline(X3000m)
qqnorm(Marathon, main ="Marathon")
qqline(Marathon)
dev.off()


# Q-Q plots of each race: considering log transform on longer races
png("a.lQQplots.png")
par(mfrow =c(2,4))
qqnorm(X100m, main ="100m")
qqline(X100m)
qqnorm(X200m, main ="200m")
qqline(X200m)
qqnorm(X400m, main ="400m")
qqline(X400m)
qqnorm(X800m, main ="800m")
qqline(X800m)
qqnorm(log(X1500m), main ="log1500m")
qqline(log(X1500m))
qqnorm(log(X3000m), main ="log3000m")
qqline(log(X3000m))
qqnorm(log(Marathon), main ="logMarathon")
qqline(log(Marathon))
dev.off()

##############################################################################
# Multivariate Normality

# Chi-Square Plot
#   requires chisplot.r in directory
source("chisplot.R")

# note: shorter distances seem normal so far. Consider upto X800m
png("chisqrShort.png")
```

```
chisplot(cbind(X100m,X200m,X400m,X800m))
dev.off()

# Pairwise normality for all shorter distance runs
# Contour plots
library(KernSmooth)
png("ContourPlots.png")
par(mfrow=c(2,3))
densSM <- bkde2D(cbind(X100m,X200m), bandwidth=c(bw.nrd(X100m),bw.nrd(X200m)))
plot(X100m,X200m)
contour(densSM$x1,densSM$x2,densSM$fhat, lty=2,nlevels=4,add=T)

densSM <- bkde2D(cbind(X100m,X400m), bandwidth=c(bw.nrd(X100m),bw.nrd(X400m)))
plot(X100m,X400m)
contour(densSM$x1,densSM$x2,densSM$fhat, lty=2,nlevels=4,add=T)

densSM <- bkde2D(cbind(X100m,X800m), bandwidth=c(bw.nrd(X100m),bw.nrd(X800m)))
plot(X100m,X800m)
contour(densSM$x1,densSM$x2,densSM$fhat, lty=2,nlevels=4,add=T)

densSM <- bkde2D(cbind(X200m,X400m), bandwidth=c(bw.nrd(X200m),bw.nrd(X400m)))
plot(X200m,X400m)
contour(densSM$x1,densSM$x2,densSM$fhat, lty=2,nlevels=4,add=T)

densSM <- bkde2D(cbind(X200m,X800m), bandwidth=c(bw.nrd(X200m),bw.nrd(X800m)))
plot(X200m,X800m)
contour(densSM$x1,densSM$x2,densSM$fhat, lty=2,nlevels=4,add=T)

densSM <- bkde2D(cbind(X400m,X800m), bandwidth=c(bw.nrd(X400m),bw.nrd(X800m)))
plot(X400m,X800m)
contour(densSM$x1,densSM$x2,densSM$fhat, lty=2,nlevels=4,add=T)
dev.off()
```

1 (b)

```
################################
# Problem 1: Part b code
################################

Data=read.table('Women.txt', header=T);
attach(Data);
X=cbind(Data[3],Data[6]);

n=length(X[,1])
p=length(X);
mu=c(colMeans(X));
Sigma=cov(X);

# Construct .50 and .95 Probability ellipse
library(ellipse)
png("P-Ellipse.png")
par(mfrow=c(1,2))
e1=ellipse(Sigma,centre=mu,level=.5)
e2=ellipse(Sigma,centre=mu,level=.95)
plot(e2, type='l',xlim=range(X[,1]),ylim=range(X[,2]))
points(X[,1],X[,2])
plot(e1, type='l',xlim=range(X[,1]),ylim=range(X[,2]))
points(X[,1],X[,2])
dev.off()

# 4-plots for Mean.
# 95% Confidence Ellipse for the mean
c2= qf(.95, p, n-p)*p*(n-1)/(n-p)
png('95CE.png')
plot(ellipse(Sigma/n, centre=mu, t=sqrt(c2)), type="l", xlab='100m',ylab='800m',main=

# 95% T^2 regions
c1=sqrt(((p*(n-1))/(n*(n-p)))*qf(.95,p,n-p)*Sigma[1,1])
c2=sqrt(((p*(n-1))/(n*(n-p)))*qf(.95,p,n-p)*Sigma[2,2])
v1=c(mu[1]-c1, mu[1] + c1)
v2=c(mu[2]-c2, mu[2] + c2)
abline(v=v1,col="blue")
abline(h=v2,col="blue")

# 95% Bonferroni Confidence
Bon=cbind(mu-qt(1-.05/(2*p), df=n-1)*sqrt(diag(Sigma)/n),
          mu + qt(1-.05/(2*p),df=n-1)*sqrt(diag(Sigma)/n))
abline(v=Bon[1,],col="red")
abline(h=Bon[2,],col="red")

# Individual 95% CI for 100m and 800m
```

```
ind1=cbind(mu[1]+qt(.5/2, df=n-1)*sqrt(Sigma[1,1]/n),
           mu[1]-qt(.5/2, df=n-1)*sqrt(Sigma[1,1]/n))

ind2=cbind(mu[2]+qt(.5/2, df=n-1)*sqrt(Sigma[2,2]/n),
           mu[2]-qt(.5/2, df=n-1)*sqrt(Sigma[2,2]/n))

abline(v=ind1, col="green")
abline(h=ind2,col="green")

# Plot sample mean
points(mu[1],mu[2])

dev.off()
```

1 (c)

```
#################################################################
# Problem 1: Part c code
#################################################################
library(xtable)
Men=read.table('Men.dat', header=T)
Women=read.table('Women.txt', header=T)

M=cbind(Men[,3:7],Men[,10])
W=cbind(Women[,3:7],Women[,9])

# Large sample size implies Anova is Robust to non-normal data
# Consider all common distances for comparison of mean

data=rbind(W,setNames(M,names(W)))
names(data)[5]="Marathon"

gender=factor(gl(2,54,108,labels=c("Women","Men")))

xtable(summary(aov(as.matrix(data[,1])~gender)))
xtable(summary(aov(as.matrix(data[,2])~gender)))
xtable(summary(aov(as.matrix(data[,3])~gender)))
xtable(summary(aov(as.matrix(data[,4])~gender)))
xtable(summary(aov(as.matrix(data[,5])~gender)))
xtable(summary(aov(as.matrix(data[,6])~gender)))
```

1 (d)

```
##############################################
# Problem1 part d
##############################################
library(xtable)

MenData=read.table('Men.dat', header=T)
attach(MenData)
Men=cbind(MenData[,3:10])
mu=colMeans(Men)
corr=cor(Men)
covv=cov(Men)

CenteredMen=Men-matrix(rep(mu,54),ncol=8,byrow=T)

# find eigenvalue/eigenvector pairs of corr matrix
Eig=eigen(corr)

# first two retain .94 percent of variance
pVar=sum(Eig$values[1:2])/sum(Eig$values)
pVar

# standardize students scores by dividing by diag elements in covv
StandardMen=t(t(CenteredMen) / diag(covv))


# biPlot of P1,P2
p1=as.matrix(StandardMen) %*% Eig$vectors[,1]
p2=as.matrix(StandardMen) %*% Eig$vectors[,2]

png("PCAbiplot.png")
plot(p1,p2, main="PCA BiPlot")
dev.off()

# order by PC1, output top ten in latex table format
x=order(-p1)
x1=p1[x]
x2=MenData$Country[x]
df=data.frame(x2,x1)
df

xtable(df[1:10,])
```

2 (a-d)

```
#########################################################################
# Problem 2 part a
#########################################################################

Ms=read.table('MS.txt',header=T)
attach(Ms)

# Seperate Groups by MS indicator
Ms0=Ms[Ms[,"MS"]==0,]
n0=length(Ms0[,1])
Ms1=Ms[Ms[,"MS"]==1,]
n1=length(Ms1[,1])

# suggested power transform for Age for MS=0
newMs0=Ms0
newMs0$Age=Ms0$Age^-0.4937005

# suggested transform: log on non zero elements MS=1
newMs1=Ms1
index=Ms1$S1diff!=0
newMs1$S1diff[index]=log(Ms1$S1diff[index])
index2=Ms1$S2diff!=0
newMs1$S2diff[index]=log(Ms1$S2diff[index2])

vars=c(2,4)
allno=Ms0[,1:5]
allyes=Ms1[,1:5]
no=Ms0[,vars]
yes=Ms1[,vars]
# mean and covariance and Spooled
mu0=colMeans(no)
mu1=colMeans(yes)

s0=cov(no)
s1=cov(yes)

Spooled=((n0-1)*s0)/(n0+n1-2)+((n1-1)*s1)/(n0+n1-2)

# Fishers discriminant function
data=rbind(no,yes)

a = as.vector(t(mu0-mu1) %*% solve(Spooled))
a
#as.matrix(a)
c1=as.matrix(no) %*% a
c2=as.matrix(yes) %*% a
```

```
m=as.numeric(a %*% (mu0 + mu1)/2)
m
# Apparent Error
(sum(c1 < m) + sum(c2 >= m))/(n0+n1)

# Classify Function used for Lachenbruch's "Holdout Procedure"
# returns binary true false as list whose index refs group
Classify <- function(pop1, pop2,i){
    ## returns # misclassification in list. index ref group
    na=nrow(pop1)
    nb=nrow(pop2)
    muA=colMeans(pop1)
    muB=colMeans(pop2)
    Sa=cov(pop1)
    Sb=cov(pop2)
    Sp=((na-1)*Sa)/(na+nb-2)+((nb-1)*Sb)/(na+nb-2)
    a=as.vector(t(muA-muB) %*% solve(Sp))
    m=as.numeric(a %*% (muA + muB)/2)
    if(i<=na){
        c1= as.matrix(pop1[i,]) %*% a
    }else{
        c2= as.matrix(pop2[i,]) %*% a
    }
    c2m=length(which( c2  >= m))
    c1m=length(which(c1 < m))
    return(c(c1m,c2m))
}

# for loop to iterate through and hold out each and classify
#   accumualte how many individuals were miscalssified.
totm=0
for(i in 1:n0+n1){

    if(i<=n0){
        index=cbind(1:n0)
        index=subset(index,index !=i)
        m1=Classify(no[index,],yes,i)
        totm= totm+m1[1]
        totm
    }else{
        index=cbind(1:n1)
        index=subset(index,index !=i)
        m2=Classify(no,yes[index,],i)
        totm=totm+m1[2]
    }
}
```

```
# Returns the Expected Actual Error Rate
EAER=totm/(n0+n1)
EAER
```