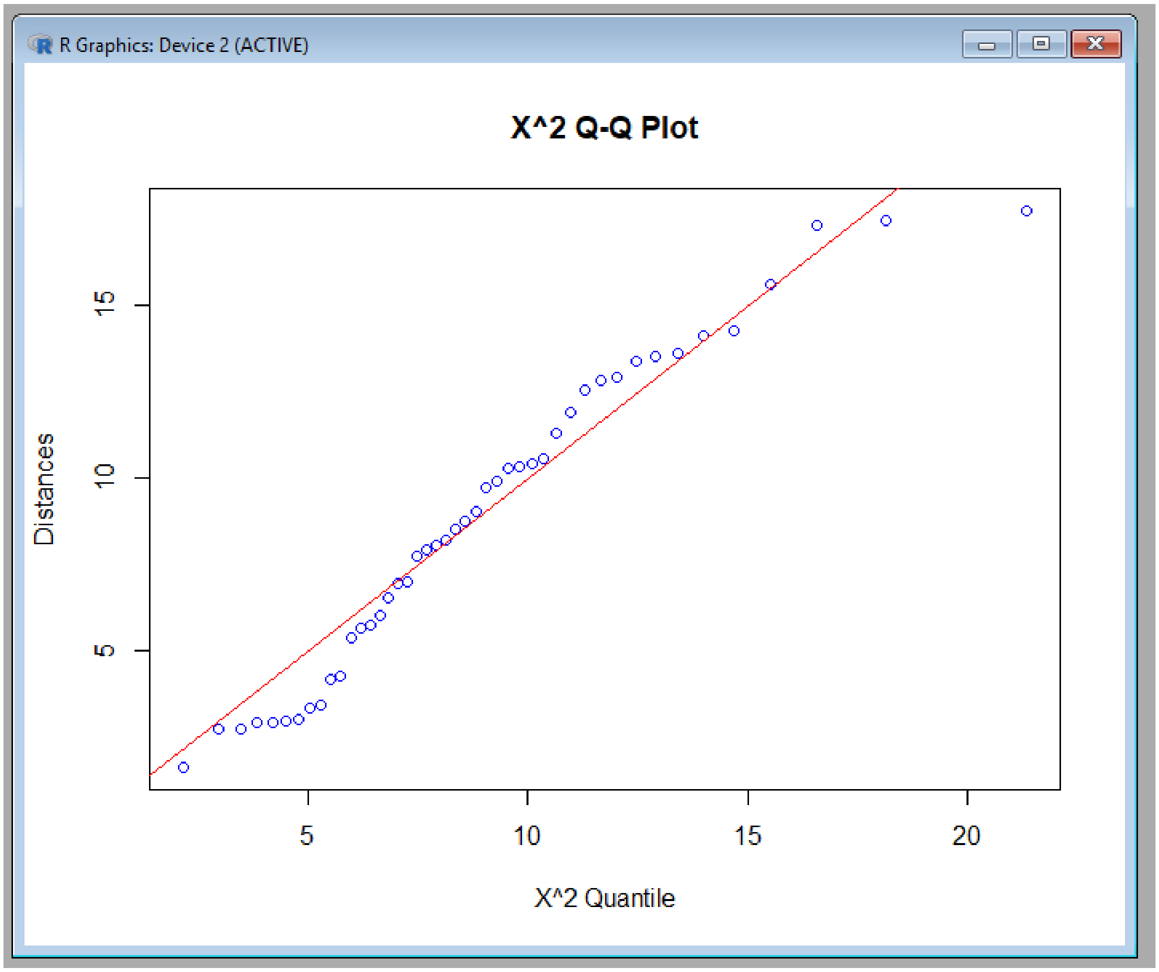
```
> pottery<-read.csv('pottery.csv')
> pottery$kiln<-factor(pottery$kiln)
> head(pottery)
  X Al2O3 Fe2O3  MgO  CaO Na2O  K2O TiO2   MnO   BaO kiln
1 1  18.8  9.52 2.00 0.79 0.40 3.20 1.01 0.077 0.015    1
2 2  16.9  7.33 1.65 0.84 0.40 3.05 0.99 0.067 0.018    1
3 3  18.2  7.64 1.82 0.77 0.40 3.07 0.98 0.087 0.014    1
4 4  16.9  7.29 1.56 0.76 0.40 3.05 1.00 0.063 0.019    1
5 5  17.8  7.24 1.83 0.92 0.43 3.12 0.93 0.061 0.019    1
6 6  18.8  7.45 2.06 0.87 0.25 3.26 0.98 0.072 0.017    1
> n<-nrow(pottery[,2:10])
> p<-ncol(pottery[,2:10])
> xmeans<- colMeans(pottery[,2:10])
> S<-cov(pottery[,2:10])
> invS<- solve(S)
> d<- apply(pottery[,2:10], 1, function(x) {t(x-xmeans) %*% invS %*% (x-xmeans)})
> layout(1)
> plot(qchisq((1:n-0.5)/n,df=p), sort(d), xlab='X^2 Quantile',
+ ylab='Distances', main='X^2 Q-Q Plot', col='blue')
> abline(a=0, b=1, col='red')
```



There is some waviness and an outlier at high X_2, but, given that 9 variables are involved, the plot is reasonably linear. The MVN appears acceptable at this stage of evaluation.

```
> man1<- manova(as.matrix(pottery[,2:10]) ~ pottery$kiln)

> summary(man1, test='Wilks')

             Df      Wilks approx F num Df den Df    Pr(>F)

pottery$kiln  4 0.00044678    23.09     36 121.66 < 2.2e-16 ***

Residuals    40

---

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> summary.aov(man1)

 Response Al2O3 :
```

```
              Df  Sum Sq Mean Sq F value    Pr(>F)
pottery$kiln  4 225.519  56.380  23.502 4.738e-10 ***
Residuals    40  95.958   2.399
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response Fe2O3 :
              Df  Sum Sq Mean Sq F value    Pr(>F)
pottery$kiln  4 234.900  58.725  118.82 < 2.2e-16 ***
Residuals    40  19.769   0.494
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response MgO :
              Df  Sum Sq Mean Sq F value    Pr(>F)
pottery$kiln  4 118.311 29.5778  77.701 < 2.2e-16 ***
Residuals    40  15.226  0.3807
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response CaO :
              Df Sum Sq Mean Sq F value    Pr(>F)
pottery$kiln  4 7.3247 1.83118  41.724 9.121e-14 ***
Residuals    40 1.7555 0.04389
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
 Response Na2O :
            Df  Sum Sq  Mean Sq F value   Pr(>F)
pottery$kiln  4 0.66651 0.166628  9.1126 2.49e-05 ***
Residuals    40 0.73141 0.018285
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response K2O :
            Df  Sum Sq Mean Sq F value    Pr(>F)
pottery$kiln  4 28.1406  7.0352  73.025 < 2.2e-16 ***
Residuals    40  3.8536  0.0963
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response TiO2 :
            Df  Sum Sq  Mean Sq F value    Pr(>F)
pottery$kiln  4 0.84324 0.210809  14.555 2.025e-07 ***
Residuals    40 0.57936 0.014484
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

 Response MnO :
            Df   Sum Sq   Mean Sq F value    Pr(>F)
pottery$kiln  4 0.077373 0.0193431  40.717 1.345e-13 ***
Residuals    40 0.019003 0.0004751
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
 Response BaO :
             Df    Sum Sq    Mean Sq F value Pr(>F)
pottery$kiln  4 0.00002602 6.5058e-06  0.7125 0.5883
Residuals    40 0.00036522 9.1305e-06
```

The 'kiln' effect on the 9 variable means is quite detectable, with a P-value < 0.0001. Clearly the mean vectors differ between at least one pair of kilns.

All of the oxides except BaO vary considerably among kilns and have P-values < 0.0001. BaO is not statistically detectable with a P-value = 0.588. BaO will probably not be useful in discriminating among kilns.

```
> require('MASS')
Loading required package: MASS
#first we use all 9 vars
> da1<- lda(kiln ~ Al2O3+Fe2O3+MgO+CaO+Na2O+K2O+TiO2+MnO+BaO , data=pottery)
> da1
Call:
lda(kiln ~ Al2O3 + Fe2O3 + MgO + CaO + Na2O + K2O + TiO2 + MnO +
    BaO, data = pottery)

Prior probabilities of groups:
         1          2          3          4          5
0.46666667 0.26666667 0.04444444 0.11111111 0.11111111

Group means:
```

```
    Al2O3    Fe2O3      MgO       CaO       Na2O      K2O      TiO2       MnO        BaO
1 16.91905 7.428571 1.842381 0.9390476 0.3457143 3.102857 0.9376190 0.07114286 0.01714286
2 12.55833 6.340000 4.931667 0.2008333 0.2550000 4.123333 0.7008333 0.12100000 0.01625000
3 11.70000 5.415000 3.855000 0.2950000 0.0500000 4.575000 0.5750000 0.09750000 0.01400000
4 18.18000 1.712000 0.674000 0.0260000 0.0540000 2.076000 1.0460000 0.00220000 0.01640000
5 17.32000 1.512000 0.606000 0.0520000 0.0480000 1.966000 0.9940000 0.00420000 0.01560000

Coefficients of linear discriminants:
              LD1        LD2         LD3          LD4
Al2O3 -0.6578047 -0.2009765  0.07950966  -0.26217266
Fe2O3  1.1057684 -1.5127138  0.25021660  -0.03241734
MgO    0.1851873  0.2357602  1.51435150  -0.21087809
CaO   -1.1569669 -3.5213496 -0.84248195   0.66015814
Na2O  -3.6519715 -3.0083348  2.49290618   0.06348730
K2O    2.3501138  2.0687213 -3.28872514  -1.46235696
TiO2  -5.9716283  2.5623191  1.94030409  -2.78050386
MnO   22.8673839 13.5688280  5.04688168  15.05722548
BaO   38.4339484 77.1899183 80.11906098 -54.39691150

Proportion of trace:
   LD1    LD2    LD3    LD4
0.6785 0.3087 0.0124 0.0004
> group<- predict(da1, method='plug-in')$class
> tab1<-table(group, pottery$kiln)
> tab1
```

```
group  1  2  3  4  5
    1 21  0  0  0  0
    2  0 12  0  0  0
    3  0  0  2  0  0
    4  0  0  0  2  1
    5  0  0  0  3  4
> tab1/nrow(pottery)*100

group         1         2         3         4         5
    1 46.666667  0.000000  0.000000  0.000000  0.000000
    2  0.000000 26.666667  0.000000  0.000000  0.000000
    3  0.000000  0.000000  4.444444  0.000000  0.000000
    4  0.000000  0.000000  0.000000  4.444444  2.222222
    5  0.000000  0.000000  0.000000  6.666667  8.888889
>
# We now exclude BaO
> da2<- lda(kiln ~ Al2O3+Fe2O3+MgO+CaO+Na2O+K2O+TiO2+MnO , data=pottery)
> da2
Call:
lda(kiln ~ Al2O3 + Fe2O3 + MgO + CaO + Na2O + K2O + TiO2 + MnO,
    data = pottery)

Prior probabilities of groups:
         1          2          3          4          5
0.46666667 0.26666667 0.04444444 0.11111111 0.11111111
```

```
Group means:
     Al2O3   Fe2O3     MgO      CaO      Na2O      K2O      TiO2       MnO
1 16.91905 7.428571 1.842381 0.9390476 0.3457143 3.102857 0.9376190 0.07114286
2 12.55833 6.340000 4.931667 0.2008333 0.2550000 4.123333 0.7008333 0.12100000
3 11.70000 5.415000 3.855000 0.2950000 0.0500000 4.575000 0.5750000 0.09750000
4 18.18000 1.712000 0.674000 0.0260000 0.0540000 2.076000 1.0460000 0.00220000
5 17.32000 1.512000 0.606000 0.0520000 0.0480000 1.966000 0.9940000 0.00420000

Coefficients of linear discriminants:
              LD1        LD2        LD3          LD4
Al2O3   0.6294327 -0.1658952 -0.1241883 -0.297631169
Fe2O3  -1.1207472 -1.5284900 -0.1906660  0.003870956
MgO    -0.1620730  0.2009679 -1.5123562 -0.233691833
CaO     0.9755527 -3.3326126  0.7147559  0.585965081
Na2O    3.4698132 -2.8548233 -2.7032389 -0.123841492
K2O    -2.3442368  2.2147791  3.1965610 -1.490392197
TiO2    5.9011976  2.6943528 -2.2643652 -3.062782627
MnO   -23.6814714 16.4914690 -8.2569338 12.910140354

Proportion of trace:
   LD1    LD2    LD3    LD4
0.6845 0.3030 0.0122 0.0004
> group2<- predict(da2, method='plug-in')$class
> tab2<-table(group2, pottery$kiln)
> tab2
```

```
group2  1  2  3  4  5

     1 21  0  0  0  0

     2  0 12  0  0  0

     3  0  0  2  0  0

     4  0  0  0  3  1

     5  0  0  0  2  4

> tab2/nrow(pottery)*100


group2         1         2         3         4         5

     1 46.666667  0.000000  0.000000  0.000000  0.000000

     2  0.000000 26.666667  0.000000  0.000000  0.000000

     3  0.000000  0.000000  4.444444  0.000000  0.000000

     4  0.000000  0.000000  0.000000  6.666667  2.222222

     5  0.000000  0.000000  0.000000  4.444444  8.888889

>
```

There is a 6.67% change of misclassifying a kiln=4 as 5, and a 2.22% chance of misclassifying a kiln=5 as a 4. These error fractions are expected to be low.

Use k-NN with k=5 to classify the data into kiln class based on all 9 oxide measurement variables. (Supply sufficient program output to indicate successful modeling.)
Find the resubstitution or a cross-validation total error for the model found in a).

```
> require('class')
Loading required package: class
> train<-scale(pottery[,2:10])
> knn5<- knn.cv(train, cl=pottery$kiln, k=5, prob=TRUE)
> knn5
[1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 5 5 5 5 4 5 5 5 4 4
attr(,"prob")
[1] 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.6 1.0 0.6
0.6 0.6 0.6 1.0 1.0 1.0 0.8 0.8
[36] 0.8 0.6 0.8 0.6 0.6 0.6 0.6 0.6 0.6 0.6
Levels: 1 2 3 4 5
> iMiss<- which(knn5!=pottery$kiln)
> sum(pottery$kiln[iMiss]=='1')
[1] 0
```

```
> sum(pottery$kiln[iMiss]=='2')
[1] 0
> sum(pottery$kiln[iMiss]=='3')
[1] 2
> sum(pottery$kiln[iMiss]=='4')
[1] 4
> sum(pottery$kiln[iMiss]=='5')
[1] 2
> length(iMiss)/nrow(pottery)*100
[1] 17.77778
# we can see from above error rate, 17.8% misclassifying rate.

# now we run knn with k=5 by exclude BaO

> require('class')

Loading required package: class

> train<-scale(pottery[,2:9])

> knn5<- knn.cv(train, cl=pottery$kiln, k=5, prob=TRUE)

> knn5

 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 5 5 5 5 4 5 5 5 4 5

attr(,"prob")

 [1] 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.8 1.0 1.0 1.0 1.0
0.8 1.0 1.0 1.0 1.0 1.0 0.8 0.8

[36] 0.8 0.6 0.8 0.6 0.6 0.6 0.6 0.6 0.6 0.6

Levels: 1 2 3 4 5

> iMiss<- which(knn5!=pottery$kiln)

> length(iMiss)/nrow(pottery)*100

[1] 15.55556

# we can see from above error rate, 15.5% misclassifying rate.

#It comparing with DA error rate less than 10%, It look DA fits better for this case.
```

In summary, The ‘kiln’ effect on the 9 variable means is quite detectable, with a P-value < 0.0001. Clearly the mean vectors differ between at least one pair of kilns.

All of the oxides except BaO vary considerably among kilns and have P-values < 0.0001. BaO is not statistically detectable with a P-value = 0.588. BaO will probably not be useful in discriminating among kilns.

So I run LDA with all 9 variables and also with 8 variables by excluding BaO. There is a 6.67% change of misclassifying a kiln=4 as 5, and a 2.22% chance of misclassifying a kiln=5 as a 4. These error fractions are expected to be low.

I also run KNN with k=5 with all variables and 8 variables by excluding BaO, the error rate is 15.5% misclassifying rate.

comparing with DA error rate less than 10%, DA fits better for this case.