MANOVA and Linear Discriminant Analysis

DS705

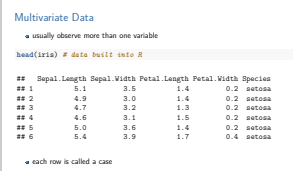
Multivariate Data

- usually observe more than one variable

```
head(iris) # data built into R
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

- each row is called a case



Multivariate Data

- usually observe more than one variable

```
head(iris) # data built into R
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

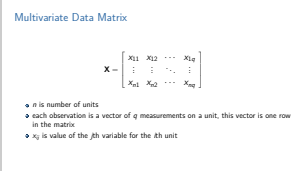
- each row is called a case

- often we observe multiple characteristics
- this yields multivariate data

Multivariate Data Matrix

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1q} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nq} \end{bmatrix}$$

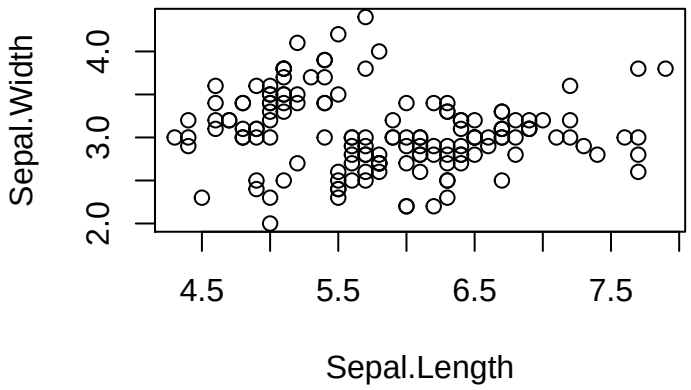
- n is number of units
- each observation is a vector of q measurements on a unit, this vector is one row in the matrix
- x_{ij} is value of the jth variable for the ith unit



- each row in the data matrix corresponds to a separate case or individual

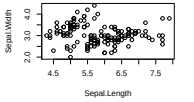
Scatterplots

```
with(iris,plot(Sepal.Length,Sepal.Width))
```



Scatterplots

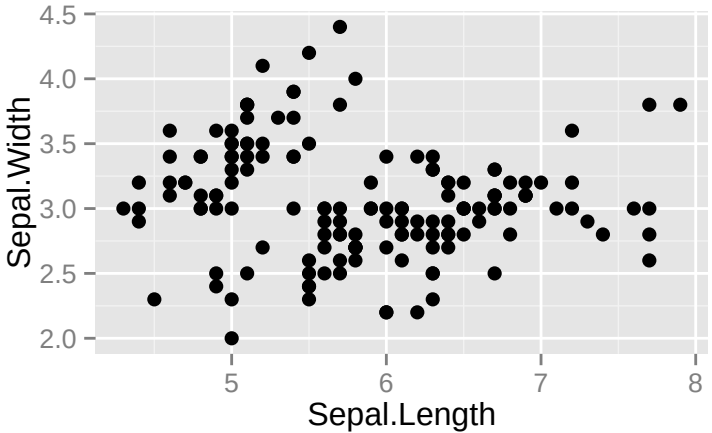
```
with(iris,plot(Sepal.Length,Sepal.Width))
```

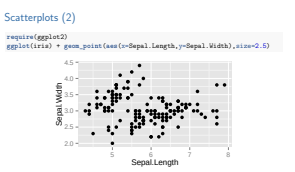


- scatterplots give us a way to explore the interactions between variables
- we can look at the variables one pair at a time as shown on this slide

Scatterplots (2)

```
require(ggplot2)
ggplot(iris) + geom_point(aes(x=Sepal.Length,y=Sepal.Width),size=2.5)
```

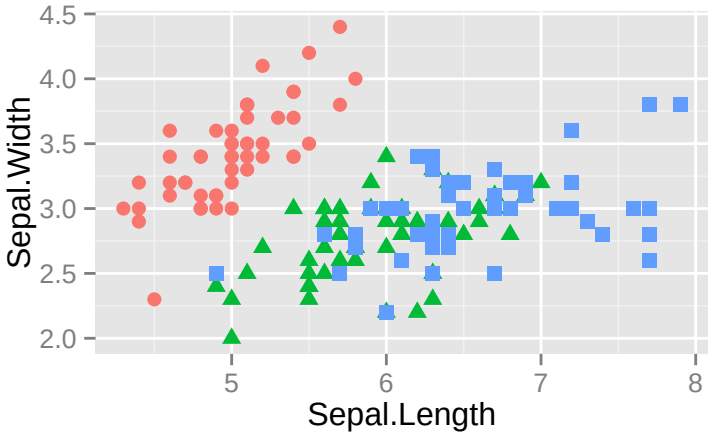


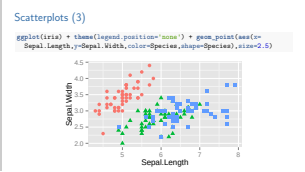


- here we see the same graph as it is produced by the ggplot package which makes nicer graphics than the base package, but ggplot takes some getting used to

Scatterplots (3)

```
ggplot(iris) + theme(legend.position='none') + geom_point(aes(x=
  Sepal.Length,y=Sepal.Width,color=Species,shape=Species),size=2.5)
```





- here is the scatterplot with the different colors showing irises of different species

Scatterplot Matrix - code

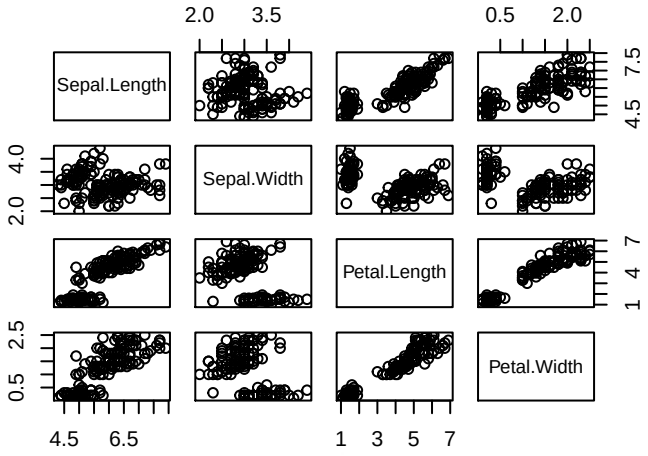
```
pairs(~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=iris)
```

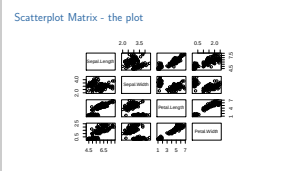
Scatterplot Matrix - code

```
pairs(~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=iris)
```

- this produces scatterplots for all the variables simultaneously
- the plot is on the next slide

Scatterplot Matrix - the plot





- note that the pairs are symmetric with graphs across the main diagonal being reflections of each other
- its tough to see much here, but we can tell all the variable pairs are positively correlated.
- try making this picture on your own in a larger window to get a better idea of how this works

Summarizing Multivariate Data

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa     :50
##  versicolor:50
##  virginica :50
```

Summarizing Multivariate Data

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa     :50
##  versicolor:50
##  virginica :50
```

- can take means, variances, etc. for each colum
- since there are four quantitative variables, think of the mean as a vector quantity containing all of the means, the median is a vector of 4 medians, etc.
- in multivariate statistics we analyze the variables together so we can account for interactions and correlations among the variables that we would miss if we treated each variable individually

Column Means

```
apply( iris[,-5], 2, mean)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

```
colMeans( iris[,-5] )
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

Column Means

```
apply( iris[,-5], 2, mean)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

```
colMeans( iris[,-5] )
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

- we can find a mean for each variable, that is, for each column separately by using the apply() command.
- notice the mean is a vector consisting of 4 sample means, one for each variable
- recall, the -5 index removes the 5th column with the categorical species variable

Column Variances

```
apply( iris[,-5], 2, var)
```

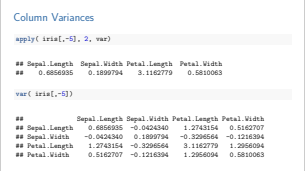
```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##    0.6856935    0.1899794    3.1162779    0.5810063
```

```
var( iris[,-5])
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
## Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
## Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
## Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
```

MANOVA and Linear Discriminant Analysis

Column Variances



Column Variances

```
apply( iris[,-5], 2, var)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##    0.6856935    0.1899794    3.1162779    0.5810063
```

```
var( iris[,-5])
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
## Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
## Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
## Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
```

- no audio

Population Covariance Matrix

$$\sigma_{ij} = \mathrm{Cov}(x_i, x_j) = E[(x_i - \mu_i)(x_j - \mu_j)]$$

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1q} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{q1} & \sigma_{q2} & \cdots & \sigma_{qq} \end{pmatrix}$$

Population Covariance Matrix

$$\sigma_{ij} = Cov(x_i, x_j) = E[(x_i - \mu_i)(x_j - \mu_j)]$$

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1q} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{q1} & \sigma_{q2} & \cdots & \sigma_{qq} \end{pmatrix}$$

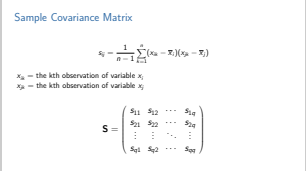
- watch out, these are variances, not standard deviations, the diagonal entries are the population variances of each variable, the notation here is standard, but a little confusing

Sample Covariance Matrix

$$s_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} (x_{ik} - \overline{x}_i)(x_{jk} - \overline{x}_j)$$

x_{ik} = the kth observation of variable x_i
x_{jk} = the kth observation of variable x_j

$$\mathbf{S} = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1q} \\ s_{21} & s_{22} & \cdots & s_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ s_{q1} & s_{q2} & \cdots & s_{qq} \end{pmatrix}$$



- the diagonal entries of the sample covariance matrix are the sample variances for each individual variable
- the off diagonal entries measure the interactions between variables and are related to the correlation
- if the variables were perfectly indepedent of each other, the off diagonal entries would be zero

Example Sample Covariance Matrix

```
x <- c(9,11,13,18,19)
y <- c(19,17,13,4,7)
sum( (x - mean(x))^2 )/(5-1)
```

```
## [1] 19
```

```
sum( (y - mean(y))^2 )/(5-1)
```

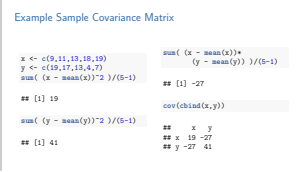
```
## [1] 41
```

```
sum( (x - mean(x))*
       (y - mean(y)) )/(5-1)
```

```
## [1] -27
```

```
cov(cbind(x,y))
```

```
##      x   y
## x   19 -27
## y -27  41
```



- if you want to see how the entries in the covariance matrix are found, study the two columns in this slide and compare the results to the cov() command

Sample Correlation Matrix

$$\text{cor}(x_i, x_j) = r_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} \frac{(x_{ik} - \overline{x}_i)}{s_i} \frac{(x_{jk} - \overline{x}_j)}{s_j}$$

x_{ik} = the kth observation of variable x_i
x_{jk} = the kth observation of variable x_j

Note: $r_{ii} = 1$

Sample Correlation Matrix

$$\text{cor}(x_i, x_j) = r_{ij} = \frac{1}{n-1} \sum_{k=1}^{n} \frac{(x_{ik} - \overline{x}_i)}{s_i} \frac{(x_{jk} - \overline{x}_j)}{s_j}$$

x_{ik} = the kth observation of variable x_i
x_{jk} = the kth observation of variable x_j
Note: $r_{ii} = 1$

- the entry in row i, column j, is the correlation coefficient between the ith variable and the jth variable.
- note that each variable is perfectly correlated with itself so the diagonal entries are 1

Example Sample Correlation Matrix

```
x <- c(9,11,13,18,19)
y <- c(19,17,13,4,7)
sx <- sd(x); mx <- mean(x)
sy <- sd(y); my <- mean(y)
1/(5-1)*sum((x-mx)^2)/(sx*sx)
```

```
## [1] 1
```

```
1/(5-1)*sum((y-my)^2)/(sy*sy)
```

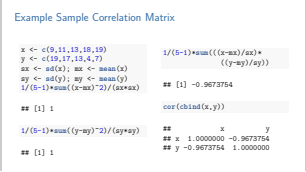
```
## [1] 1
```

```
1/(5-1)*sum(((x-mx)/sx)*
              ((y-my)/sy))
```

```
## [1] -0.9673754
```

```
cor(cbind(x,y))
```

```
##            x          y
## x  1.0000000 -0.9673754
## y -0.9673754  1.0000000
```



- this slide walks through an example of finding the correlation matrix for two quantitative variables x and y
- we don't have to do all these calculations, rather they show us what goes into the matrix
- compare the result to that produced by the cor() command

Compare Covariance and Correlation

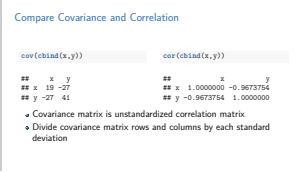
```
cov(cbind(x,y))
```

```
##     x   y
## x  19 -27
## y -27  41
```

```
cor(cbind(x,y))
```

```
##            x          y
## x  1.0000000 -0.9673754
## y -0.9673754  1.0000000
```

- Covariance matrix is unstandardized correlation matrix
- Divide covariance matrix rows and columns by each standard deviation


Compare Covariance and Correlation

```
cov(cbind(x,y))
```

```
##    x   y
## x 19 -27
## y -27  41
```

```
cor(cbind(x,y))
```

```
##            x          y
## x  1.0000000 -0.9673754
## y -0.9673754  1.0000000
```

- Covariance matrix is unstandardized correlation matrix
- Divide covariance matrix rows and columns by each standard deviation

- you can find the correlation matrix by starting with covariance matrix and computing individual standard deviations as the square root of the variances along the diagonal
- then divide the first row and column by the first standard deviation, the second row and column by the second standard deviation, etc.

Multivariate Normal Distribution

$$\mathbf{x} \sim N\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

- $\mathbf{x}$ is a vector of q numbers
- $\boldsymbol{\mu}$ is the population mean vector of length q
- $\boldsymbol{\Sigma}$ is the $q \times q$ population covariance matrix

Multivariate Normal Distribution

$$\mathbf{x} \sim N\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$$

- $\mathbf{x}$ is a vector of q numbers
- $\boldsymbol{\mu}$ is the population mean vector of length q
- $\boldsymbol{\Sigma}$ is the $q \times q$ population covariance matrix

- for a multivariate normal distribution we specify a vector of means and the population covariance matrix
- a matrix is required since we are specifying how the variables interact

Example

```
require(MASS)
mu <- c(12,30); Sigma <- rbind( c(.8,.5),c(.5,2) )
x <- mvrnorm(1000,mu,Sigma)
plot(x[,1],x[,2],xlim=c(8,16),ylim=c(26,34))
ellipse(mu,Sigma,sqrt(qchisq(.5,2)),col='blue')
ellipse(mu,Sigma,sqrt(qchisq(.95,2)),col='blue')
```

- Plot on next page.

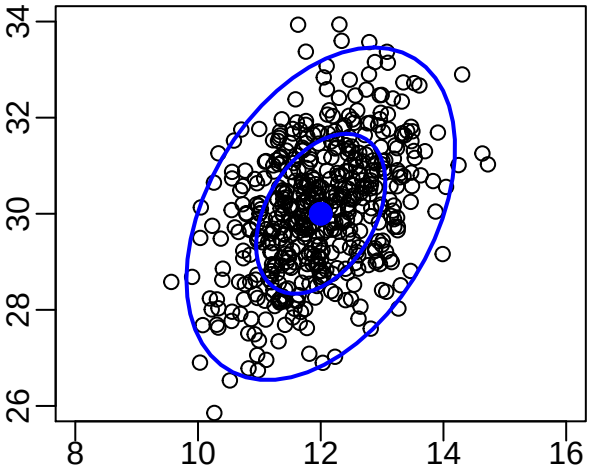

Example

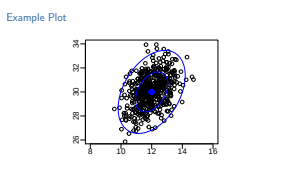
```
require(MASS)
mu <- c(12,30); Sigma <- rbind( c(.8,.5),c(.5,2) )
x <- mvrnorm(1000,mu,Sigma)
plot(x[,1],x[,2],xlim=c(8,16),ylim=c(26,34))
ellipse(mu,Sigma,sqrt(qchisq(.5,2)),col='blue')
ellipse(mu,Sigma,sqrt(qchisq(.95,2)),col='blue')
```

- Plot on next page.

- here we show how to generate random numbers from a multivariate distribution
- the added code produces some ellipses that should contain approximately 50% and 95% of the observations
- the plot is shown on the next page

Example Plot





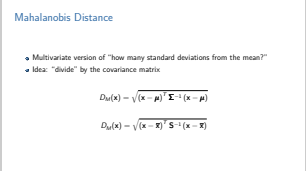
- the ellipses shown here are contours that contain about 50% of the observations and 95% of the observations
- the shape of the ellipse is determined by the covariance and the radius the ellipse is determined by a chi-square distribution
- we haven't studied chi-square distributions much, though we've used them, a variable with a chi-square distribution is a sum of independent squared standard normal random variables
- don't worry too much about the details of chi-squares, the main thing is that the squared distance from the mean of an observation in a multivariate normal essentially follows a chi-square distribution

Mahalanobis Distance

- Multivariate version of "how many standard deviations from the mean?"
- Idea: "divide" by the covariance matrix

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}$$

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \overline{\mathbf{x}})^T \mathbf{S}^{-1} (\mathbf{x} - \overline{\mathbf{x}})}$$



- because the covariance is given by a matrix division means multiplying by the inverse of the matrix which is a linear algebra idea
- fortunately we can still work with this distance in R even if we don't know the linear algebra

Mahalanobis Example

```
mu <- c(12,30); Sigma <- rbind( c(.8,.5),c(.5,2) )
x <- mvrnorm(500,mu,Sigma)
dsquare <- mahalanobis(x,mu,Sigma)
hist(dsquare,prob=TRUE,breaks=30)
curve(dchisq(x, df=2),
          col="darkblue", lwd=2, add=TRUE, yaxt="n")
```

Plot on next slide.

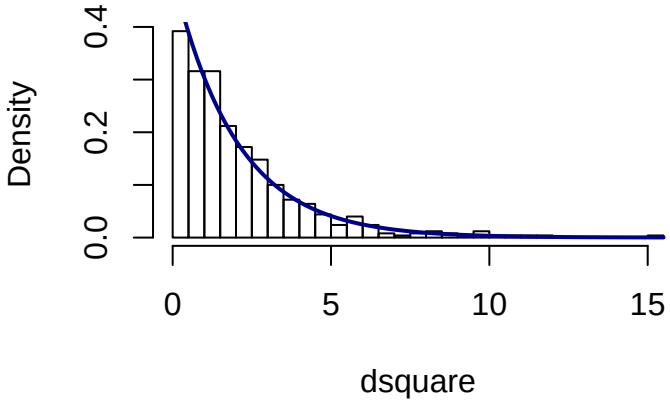
Mahalanobis Example

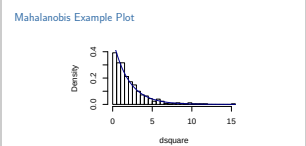
```
mu <- c(12,30); Sigma <- rbind( c(.8,.5),c(.5,2) )
x <- mvrnorm(500,mu,Sigma)
dsquare <- mahalanobis(x,mu,Sigma)
hist(dsquare,prob=TRUE,breaks=30)
curve(dchisq(x, df=2),
          col="darkblue", lwd=2, add=TRUE, yaxt="n")
```

Plot on next slide.

- computing Mahalanobis distance is the multivariate version of the z-score
- when we compute z-scores we expect the data from a normal distribution to follow a standard normal distriubtion
- when we compute Mahalanobis distance, we expect the squared distances to follow a chi-square distribution
- the plot is on the next slide

Mahalanobis Example Plot

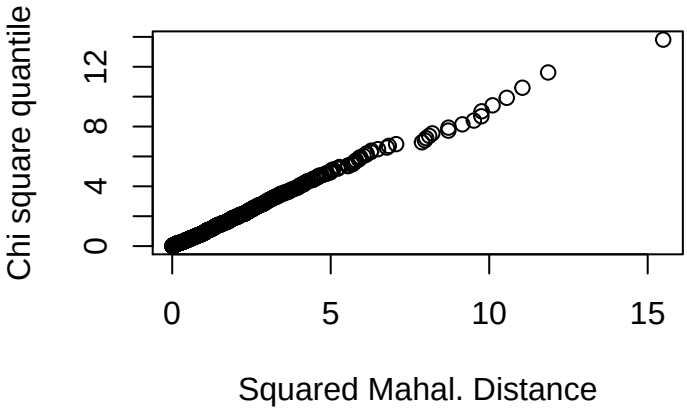




- here we are showing a histogram of the squared mahalanobis distances for our multivariate normal random sample
- also plotted for comparison is the density curve of a chi-square distribution with 2 degrees of freedom for comparision
- the degrees of freedom is the same as the number of dimensions in each observation, here our multivariate normal is two dimensional

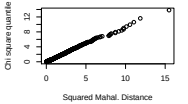
Chi square quantile plot

```
plot(sort(dsquare),qchisq(ppoints(500),df=2),
     xlab='Squared Mahal. Distance', ylab='Chi square quantile')
```



Chi square quantile plot

```
plot(sort(dsquare),qchisq(ppoints(500),df=2),
     xlab='Squared Mahal. Distance', ylab='Chi square quantile')
```



- by comparing the squared distances to the mean to the theoretical quantiles from a chi-square distribution we get a kind of normal probability plot for assessing multivariate normality
- we will see a simpler way to produce a chi square quantile plot in a few slides

Assessing Multivariate Normality

- Chi square quantile plot $\rightarrow$ want a straight line
- no *best* hypothesis test
- MVN paackage
  - Henze-Zinkler - `hzTest()`
  - Royston - `roystonTest()`
  - Mardia - `mardiaTest()`
- try all three
  - good agreement $\Rightarrow$ stop
  - marginal significance or inconsistent results $\Rightarrow$ look harder
- beware of small samples

Assessing Multivariate Normality

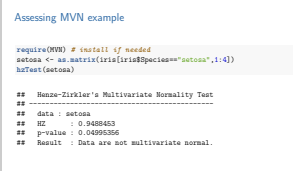
- Chi square quantile plot → want a straight line
- *no best* hypothesis test
- MVN paackage
  - Henze-Zinkler - `hzTest()`
  - Royston - `roystonTest()`
  - Mardia - `mardiaTest()`
- try all three
  - good agreement ⇒ stop
  - marginal significance or inconsistent results ⇒ look harder
- beware of small samples

- to assess multivariate normality
- start with a chi square quantile plot and look for something that is approximately linear, only a strong systematic deviation from linear should lead to rejecting normality
- for small samples, allow a lot of latitude before rejecting normality

Assessing MVN example

```r
require(MVN) # install if needed
setosa <- as.matrix(iris[iris$Species=="setosa",1:4])
hzTest(setosa)
```

```
##   Henze-Zirkler's Multivariate Normality Test
## ---------------------------------------------
##   data : setosa
##   HZ      : 0.9488453
##   p-value : 0.04995356
##   Result  : Data are not multivariate normal.
```


Assessing MVN example

```
require(MVN) # install if needed
setosa <- as.matrix(iris[iris$Species=="setosa",1:4])
hzTest(setosa)
```

```
##   Henze-Zirkler's Multivariate Normality Test
## ---------------------------------------------
##   data : setosa
##   HZ      : 0.9488453
##   p-value : 0.04995356
##   Result  : Data are not multivariate normal.
```

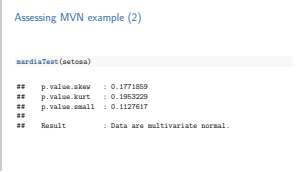
- we'll start with the usual hypothesis tests of normality
- the null hypothesis the that the distribution is multivariate normal
- the alternative is that the distribution is NOT multivariate noramal
- frequently you may find that the hypothesis tests are not completely in agreement
- the Henze Zinkler test just barely rejects normality

Assessing MVN example (2)

```
mardiaTest(setosa)
```

```
##     p.value.skew    : 0.1771859
##     p.value.kurt    : 0.1953229
##     p.value.small   : 0.1127617
##
##     Result          : Data are multivariate normal.
```

Assessing MVN example (2)



- the Mardia test really tests for two things separately
- it tests for skewness (lack of symmetry)
- it also tests for kurtosis which essentially is asking if the tails decay at the right rate
- the p.value.small is another version of the skewness p-value that includes a small sample correction
- in this case, all of the p-values are large giving us no reason to reject normality

Assessing MVN example (3)

```
roystonTest(setosa)
```

```
##   Royston's Multivariate Normality Test
## ---------------------------------------------
##   data : setosa
##
##   H       : 31.51803
##   p-value : 2.187653e-06
##
##   Result  : Data are not multivariate normal.
```

Assessing MVN example (3)

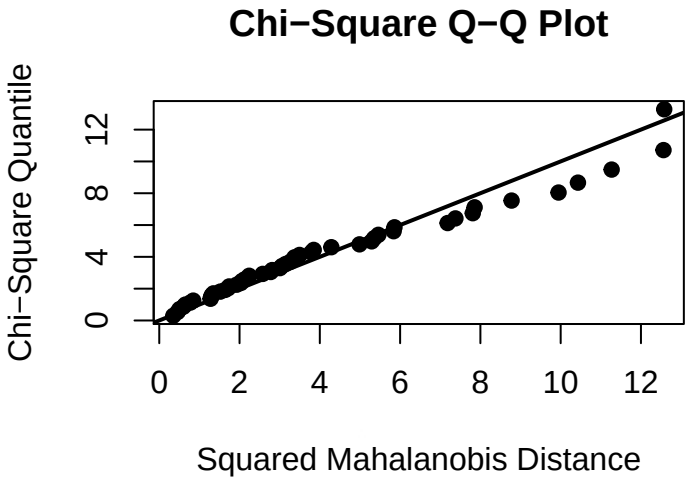
```
royston Test(setosa)
```

```
##   Royston's Multivariate Normality Test
## ---------------------------------------------
##   data : setosa
##
##   H        : 31.51803
##   p-value : 2.187653e-06
##
##   Result   : Data are not multivariate normal.
```

- the Royston test strongly rejects normality because of the very small P-value
- so now we have three different hypothesis test that essentially said not sure, yes, no, so now what?

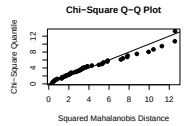
Assessing MVN example (4)

- tests ambiguous so `hzTest(setosa,qqplot=TRUE)`



Assessing MVN example (4)

- tests ambiguous so hzTest(setosa,qqplot=TRUE)

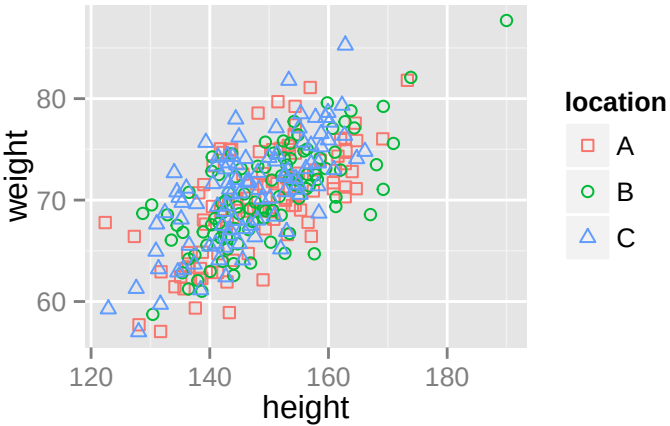


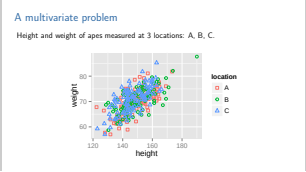

```
##   Henze-Zirkler's Multivariate Normality Test
```

- the simplest way to make a normal quantile plot is set the qqplot variable to true and call one of the multivariate normality tests
- at the upper end of the chi-square quantile plot the observations deviate significantly from the expected linear trend
- since the points lie below the line this indicates that the observations are not as far from the mean as they should be, that is, the distribution has light tails
- since there is a distinct pattern in the deviation from linear, combined with the normality tests, we reject normality for this data

A multivariate problem

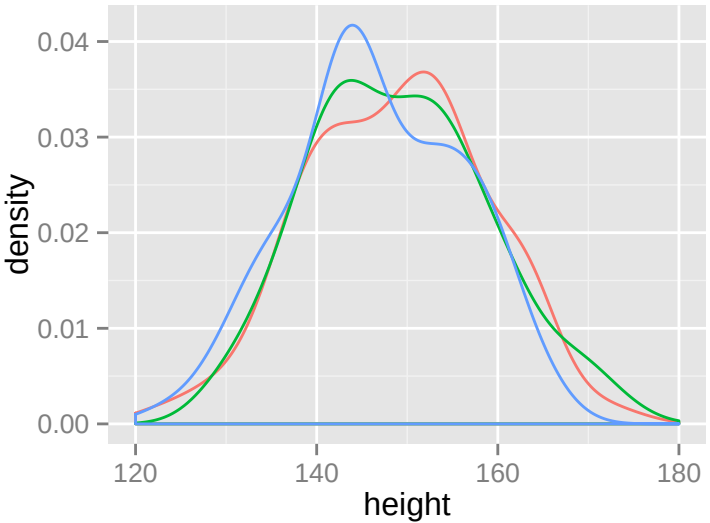
Height and weight of apes measured at 3 locations: A, B, C.



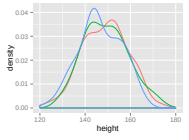


- a biologist has collected height and weight data for 100 random apes at each of 3 locations
- are there significant differences between the population mean vectors of height and weight for these three populations of apes?
- we can, of course look at height and weight separately, but since there is a strong correlation between height and weight it may not make sense to look at these variables separately
- over the next few slides we'll look at what happens if we do analyze height and weight separately

Different mean heights?



Different mean heights?



- shown here are approximated density curves that are fit to the sample height data for each of the three groups, they are all centered very close together so it will be difficult to separate the groups
- note that the distributions look roughly normal and all have the same variance, so a one-variable anova would be appropriate to analyze the height.

ANOVA on heights

```
aov.model <- aov(height~location,data=apes)
summary(aov.model)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## location      2    423  211.74   2.125  0.121
## Residuals   297  29599   99.66
```

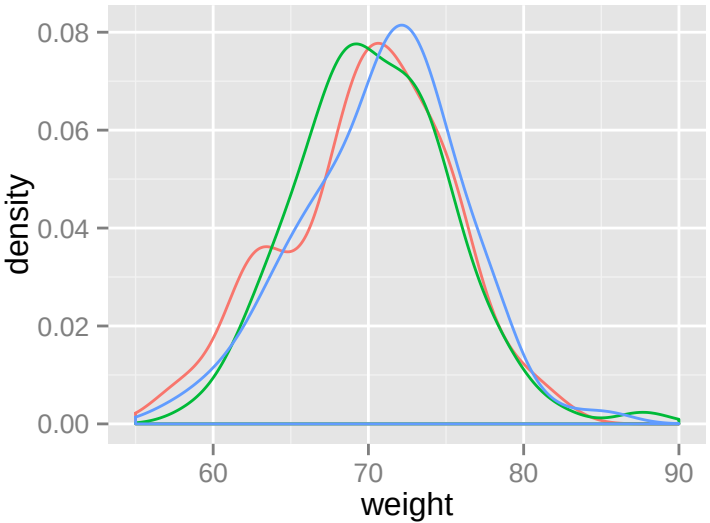
ANOVA on heights

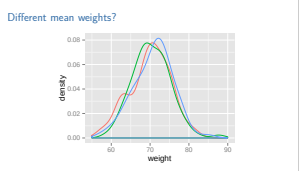
```
aov.model <- aov(height~location,data=apes)
summary(aov.model)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## location      2    423  211.74   2.125  0.121
## Residuals   297  29599   99.66
```

- here we show the results of conducting the oneway ANOVA to test for differences in the population mean heights
- the large P-value, .121, indicates that there are not statistically significant differences in the population mean heights

Different mean weights?





- shown here are approximated density curves that are fit to the sample data for each of the three groups, they are all centered very close together so it will be difficult to separate the groups
- note that the distributions look roughly normal and all have the same variance, so a one-variable anova would be appropriate to analyze the height.

ANOVA on weights

```
aov.model <- aov(weight~location,data=apes)
summary(aov.model)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## location      2     38   19.18   0.738  0.479
## Residuals   297   7719   25.99
```

ANOVA on weights

```
aov.model <- aov(weight~location,data=apes)
summary(aov.model)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## location      2     38   19.18   0.738  0.479
## Residuals   297   7719   25.99
```

- again the large P value, .479, indicates there are not significant differences in the population mean weights

Why not multiple ANOVA?

- univariate analysis of each variable misses correlations
- multiple tests requires correction to maintain FWER so power is lost

Why not multiple ANOVA?
- univariate analysis of each variable misses correlations
- multiple tests requires correction to maintain FWER so power is lost

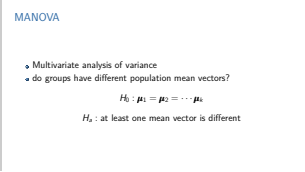
- doing multiple single variable ANOVAs doesn't account for correlations between variables.
- also, if doing multiple ANOVA, we have the usual multiple comparisons problem and have to take care to ensure that the family wise error rate is preserved. A Bonferroni correction could be applied for instance.

MANOVA

- Multivariate analysis of variance
- do groups have different population mean vectors?

$$H_0 : \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \cdots \boldsymbol{\mu}_k$$

H_a : at least one mean vector is different



- Multivariate Analysis of Variance, or MANOVA, tests to see if there are statistically significant differences among the population mean vectors
- in the MANOVA context, the independent variable is the factor or group variable and the dependent variables are the those that might depend on the group such as height and weight.
- the analysis takes into account correlations among the dependent variables, such as weight and height

MANOVA is not always appropriate

- if dependendent variables are uncorrelated, then use ANOVA on each variable and correct for multiple tests
- if dependent variables are multicollinear, then should eliminate redundant variables before trying MANOVA

MANOVA is not always appropriate

- if dependendent variables are uncorrelated, then use ANOVA on each variable and correct for multiple tests
- if dependent variables are multicollinear, then should eliminate redundant variables before trying MANOVA

- if the dependent variables are uncorrelated then use multiple single variable ANOVAs and correct for multiple tests
- if there are redundancies among the variables then the redundant variables should be removed before running ANOVA, this can be done by looking for variables with very large Variance Inflaction Factors and eliminating those variables as is done in multiple linear regression

MANOVA requirements

- independent random samples from each population
- data is from multivariate normal distributions
- each distribution has the same covariance matrix

MANOVA requirements

- independent random samples from each population
- data is from multivariate normal distributions
- each distribution has the same covariance matrix

- these requirements are the multavriate versions of the requirements for one variable ANOVA

MANOVA Idea

Like ANOVA

$$\text{test stat} \approx \frac{\text{covariance between groups}}{\text{covariance within groups}}$$

but,

- the (co)variances are now matrices
- at least four ways to compute a test statistic

MANOVA Idea

Like ANOVA

$$\text{test stat} \propto \frac{\text{covariance between groups}}{\text{covariance within groups}}$$

but,

- the (co)variances are now matrices
- at least four ways to compute a test statistic

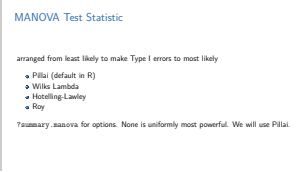
- if you want to know about the matrices involved here, there are plenty of textbooks and online resources to find that information. A background in linear algebra is helpful.
- because the test statistic is a matrix, there are multiple ways to compute a single test statistic from the matrix, all the different ways result in something which has approximately an F distribution like in ANOVA.

MANOVA Test Statistic

arranged from least likely to make Type I errors to most likely

- Pillai (default in R)
- Wilks Lambda
- Hotelling-Lawley
- Roy

`?summary.manova` for options. None is uniformly most powerful. We will use Pillai.



- the Pillai test statistic is the most conservative which means less likely to make type I errors, but also generally less powerful
- the Roy test statistic is the most liberal which means more type I errors, but generally more powerful

MANOVA Example (1)

Are the apes at locations A, B, and C different in terms of mean height and weight? (different mean vectors?)

$$H_0 : \boldsymbol{\mu}_A = \boldsymbol{\mu}_B = \boldsymbol{\mu}_C$$

H_a : at least one mean vector is different

MANOVA Example (1)

Are the apes at locations A, B, and C different in terms of mean height and weight? (different mean vectors?)

$H_0 : \boldsymbol{\mu}_A = \boldsymbol{\mu}_B = \boldsymbol{\mu}_C$

H_a : at least one mean vector is different

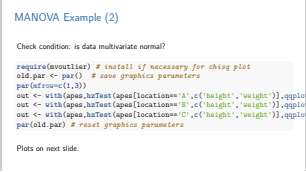
- just like ANOVA, MANOVA is an omnibus test. it can tell us there are significant differences, but it can't tell what those differences are
- first we'll check the conditions for MANOVA

MANOVA Example (2)

Check condition: is data multivariate normal?

```
require(mvoutlier) # install if necessary for chisq plot
old.par <- par()  # save graphics parameters
par(mfrow=c(1,3))
out <- with(apes,hzTest(apes[location=='A',c('height','weight')],qqplo
out <- with(apes,hzTest(apes[location=='B',c('height','weight')],qqplo
out <- with(apes,hzTest(apes[location=='C',c('height','weight')],qqplo
par(old.par) # reset graphics parameters
```

Plots on next slide.


MANOVA Example (2)

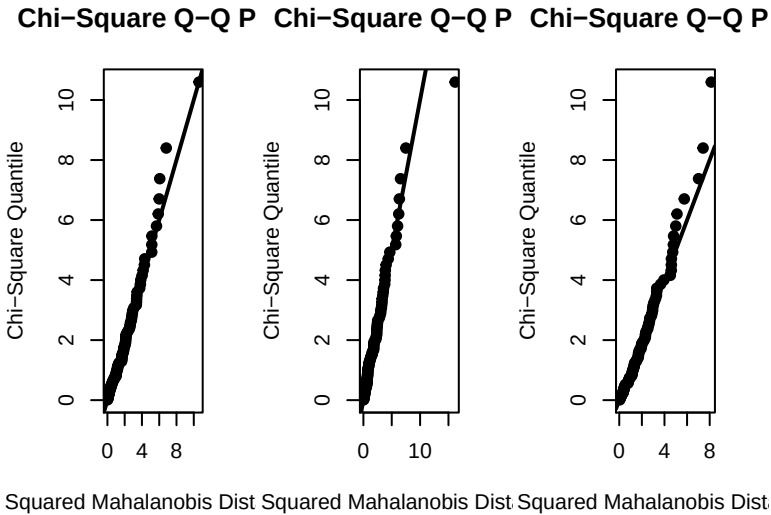
Check condition: is data multivariate normal?

```
require(mvoutlier) # install if necessary for chisq plot
old.par <- par()  # save graphics parameters
par(mfrow=c(1,3))
out <- with(apes,hzTest(apes[location=='A',c('height','weight')],qqplot=TRUE))
out <- with(apes,hzTest(apes[location=='B',c('height','weight')],qqplot=TRUE))
out <- with(apes,hzTest(apes[location=='C',c('height','weight')],qqplot=TRUE))
par(old.par) # reset graphics parameters
```

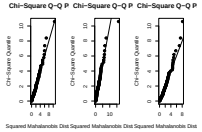
Plots on next slide.

- there is a little bit of funny business here to supress the output of the hzTest and show only the plots, you can mimic this in your homework, but change eval=FALSE to echo=FALSE

MANOVA Example (3)



MANOVA Example (3)



- the chi square quantile plots are reasonably linear suggesting the three samples come from multivariate distributions

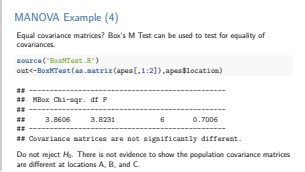
MANOVA Example (4)

Equal covariance matrices? Box's M Test can be used to test for equality of covariances.

```
source('BoxMTest.R')
out<-BoxMTest(as.matrix(apes[,1:2]),apes$location)
```

```
## ------------------------------------------------
##  MBox Chi-sqr. df P
## ------------------------------------------------
##     3.8606     3.8231           6       0.7006
## ------------------------------------------------
## Covariance matrices are not significantly different.
```

Do not reject H_0. There is not evidence to show the population covariance matrices are different at locations A, B, and C.


MANOVA Example (4)

Equal covariance matrices? Box's M Test can be used to test for equality of covariances.

```
source('BoxMTest.R')
out<-BoxMTest(as.matrix(apes[,1:2]),apes$location)
```

```
## ----------------------------------------------------
##  MBox Chi-sqr. df P
## ----------------------------------------------------
##     3.8606     3.8231          6        0.7006
## ----------------------------------------------------
## Covariance matrices are not significantly different.
```

Do not reject H_0. There is not evidence to show the population covariance matrices are different at locations A, B, and C.

- Box's M test isn't readily available in an R package that we could find, but the source code is in the download packet and you can include it as shown above, make sure it is in your working directory
- equal covariances and multivariate normals means we're ready to apply MANOVA

MANOVA Example (5)

```
lmodel <- lm(cbind(height,weight)~location,data=apes)
m.out <- manova(lmodel)
summary(m.out,test="Pillai")
```

```
##             Df   Pillai approx F num Df den Df   Pr(>F)
## location     2 0.050308   3.8318      4    594 0.004384 **
## Residuals 297
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Reject H_0. There is strong evidence to show apes at the three locations are different in terms of population mean weight and height.

MANOVA Example (5)

```
lmodel <- lm(cbind(height,weight)~location,data=apes)
m.out <- manova(lmodel)
summary(m.out,test="Pillai")
```

```
##            Df   Pillai approx F num Df den Df   Pr(>F)
## location    2 0.050308   3.8318      4    594 0.004384 **
## Residuals 297
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Reject H_0. There is strong evidence to show apes at the three locations are different in terms of population mean weight and height.

- the MANOVA analysis shows that there are clear differences between groups of Apes for the population mean vectors when height and weight are considered together
- now we'll try to answer the question - which groups are significantly different?

Posthoc Analysis

Often follow up with univariate ANOVAs. Shortcut:

```
summary.aov(m.out)
```

```
##  Response height :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## location      2   423.5 211.742  2.1247 0.1213
## Residuals   297 29598.5  99.658
##
##  Response weight :
##              Df Sum Sq Mean Sq F value Pr(>F)
## location      2   38.4   19.18   0.738 0.4789
## Residuals   297 7719.0   25.99
```

Posthoc Analysis

Often follow up with univariate ANOVAs. Shortcut:

```
summary.aov(m.out)
```

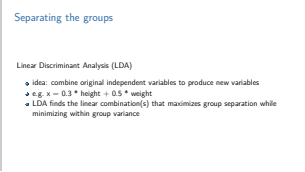
```
##  Response height :
##              Df  Sum Sq Mean Sq F value Pr(>F)
## location      2   423.5 211.742  2.1247 0.1213
## Residuals   297 29598.5  99.658
##
##  Response weight :
##              Df Sum Sq Mean Sq F value Pr(>F)
## location      2   38.4   19.18   0.738 0.4789
## Residuals   297 7719.0   25.99
```

- some statisticians say that the univariate ANOVAs are 'protected' by the MANOVA if you choose a conservative test statistic such as Pillai or Wilks Lambda
- protected means that it isn't necessary to correct for multiple comparisons when running multiple ANOVA
- however, strictly speaking there is no protection unless the null is totally true or the alternative is totally true, so we should apply a bonferroni correction and use alpha / 2 for each univariate ANOVA
- in this case, it doesn't matter, the univariate ANOVAs suggest that are no significant differences in population mean weights or heights when considered separately

Separating the groups

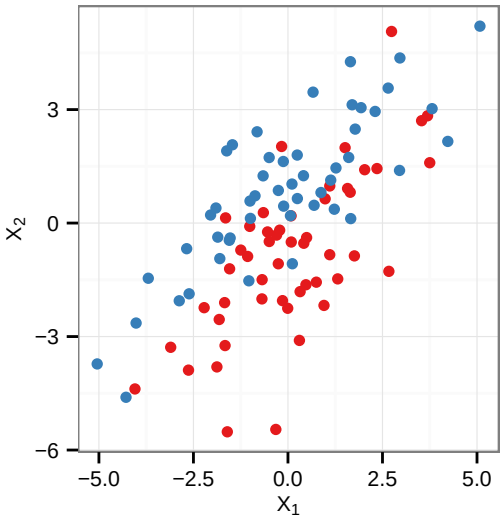
Linear Discriminant Analysis (LDA)

- idea: combine original independent variables to produce new variables
- e.g. x = 0.3 * height + 0.5 * weight
- LDA finds the linear combination(s) that maximizes group separation while minimizing within group variance

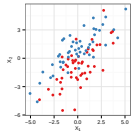


- the univariate ANOVA's cannot capture the relationships among the dependent variables
- Linear Discriminant Analysis attempts to find a combination of the variables which maximizes the separation between groups
- the underlying computation is based on eigenvectors of a matrix, but we can see how to use the tools in R
- Discriminant Analysis is primarily used as a means of classifying new observations into groups and comes up in machine learning, but it is also useful here.

LDA - picture 1

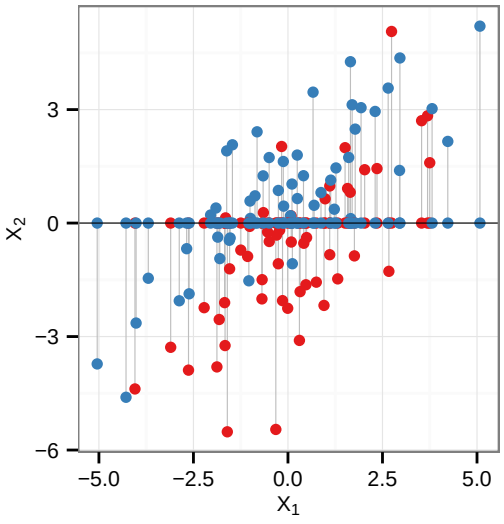


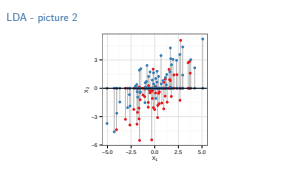
LDA - picture 1



- how does linear discriminant analysis works
- imagine we two multivariate data sets, the red and the blue
- while they overlap, they clearly seem to have different centers

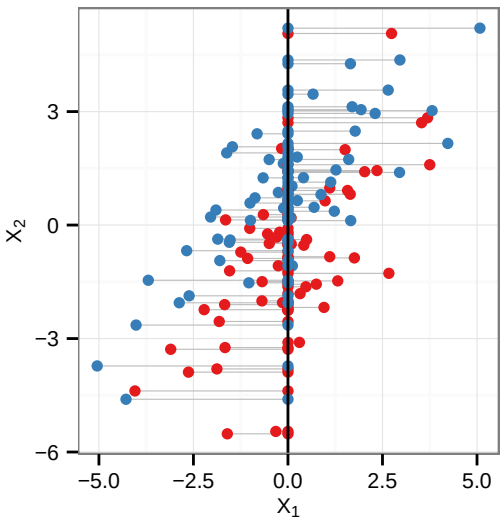
LDA - picture 2

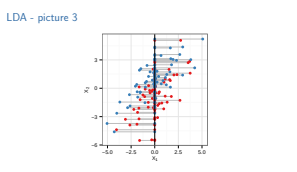




- this picture shows what happens when we consider only the first variable x1
- you can think of this as projecting the data onto the horizontal or x1 axis
- notice how the red and blue data are still intermingled and overlapping when projected onto the x1 axis

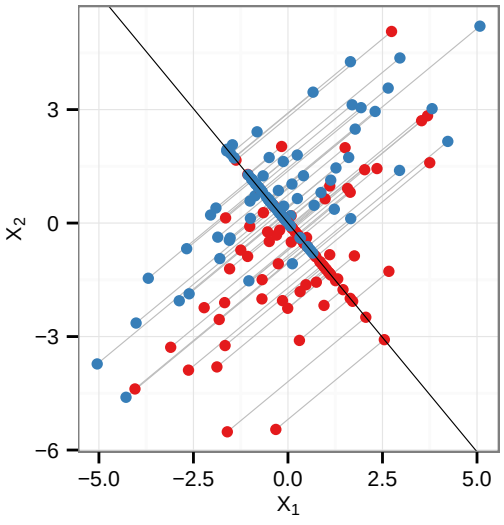
LDA - picture 3

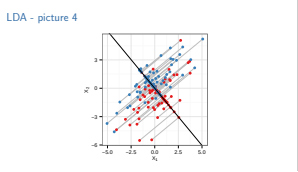




- this picture shows what happens when we consider only the second variable x2
- we are projecting the data onto the vertical or x2 axis
- again the red and blue data are still intermingled and overlapping

LDA - picture 4





- Linear Discriminant Analysis forms linear combinations of the original variables that maximize the separation between groups while minimizing the within group variance
- Think of it as choosing a new axis so that when the data is projected onto the new axis the separation is maximized
- Notice how the red and blue data is less mingled and less overlapping on the new axis
- we can test the new combination of variables for differences between the groups

How to separate groups

- use linear discriminant analysis to find combination of variables that maximizes group separation
- apply univariate multiple comparison procedure to new variable

How to separate groups

- use linear discriminant analysis to find combination of variables that maximizes group separation
- apply univariate multiple comparison procedure to new variable

- no audio

Separating Groups of Apes

Here is the R code:

```
fit <- lda( location~height+weight,data=apes) # fit model
plda <- predict(object=fit, newdata=apes) # compute combinations
ld1 <- plda$x[,1] # extract most separating combination
ld2 <- plda$x[,2] # second most separating combination
```

- linear discriminant analysis actually produces multiple separating directions, one fewer than the number of groups, these separating directions are used to classify new data into groups
- we'll focus on the first linear combination, ld1, this direction accounts for the largest proportion of the total separation among the groups

New variables scatterplot - code

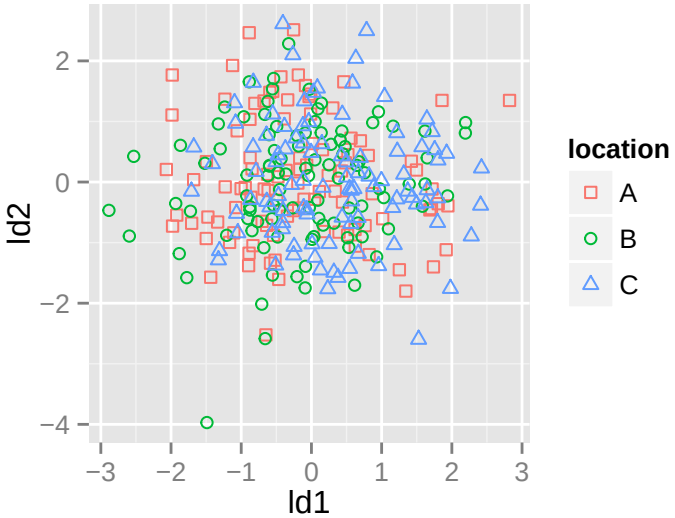
```
apes <- data.frame(apes,ld1=ld1,ld2=ld2)
ggplot(apes)
  + geom_point( aes( x=ld1, y=ld2, color=location, shape=location) )
  + scale_shape_manual( values=c(0,1,2) )
  + scale_size_manual( values=2*c(1,1,1) )
```

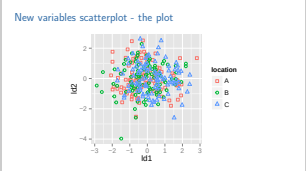
New variables scatterplot - code

```
apes <- data.frame(apes,ld1=ld1,ld2=ld2)
ggplot(apes)
  + geom_point( aes( x=ld1, y=ld2, color=location, shape=location) )
  + scale_shape_manual( values=c(0,1,2) )
  + scale_size_manual( values=2*c(1,1,1) )
```

- here we use the new variables ld1 and ld2 as axes in a scatterplot
- the code is shown here and the scatterplot is shown on the next slide

New variables scatterplot - the plot

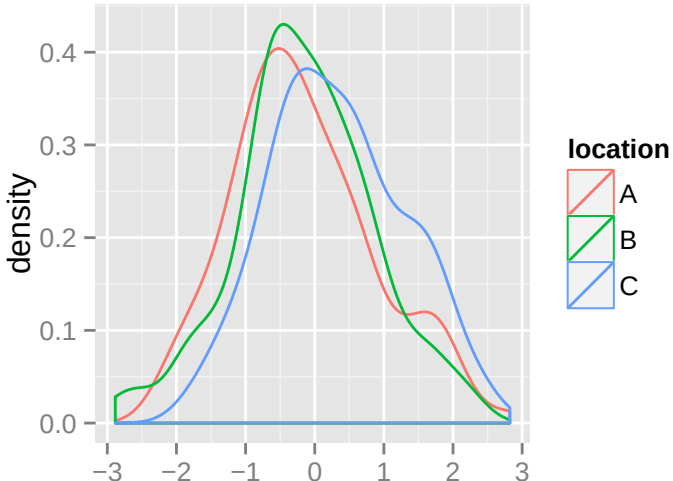


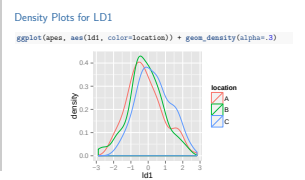


- the groups aren't exactly distinctly separated, but notice that the blue group, from location C seems to be mostly to the right, while the red and green groups from locations A and B seem to be mostly to the left

Density Plots for LD1

```
ggplot(apes, aes(ld1, color=location)) + geom_density(alpha=.3)
```





- here are estimated density curves for the variable ld1 at locations A, B, and C,
- notice how the density curve for group C appears shifted to the right of the density curves for the other groups
- compare this to the density curves for height and weight that we showed near the beginning of the presentation

Apply Tukey to LD1

```
linear.model<-aov(ld1~location,data=apes)
TukeyHSD(linear.model)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = ld1 ~ location, data = apes)
##
## $location
##             diff        lwr       upr     p adj
## B-A 0.009560405 -0.3235611 0.3426819 0.9974836
## C-A 0.485246799  0.1521253 0.8183683 0.0019781
## C-B 0.475686394  0.1425649 0.8088079 0.0025002
```

Apply Tukey to LD1

```
linear.model<-aov(ld1~location,data=apes)
TukeyHSD(linear.model)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = ld1 ~ location, data = apes)
##
## $location
##           diff        lwr       upr     p adj
## B-A 0.009560405 -0.3235611 0.3426819 0.9974836
## C-A 0.485246799  0.1521253 0.8183683 0.0019781
## C-B 0.475686394  0.1425649 0.8088079 0.0025002
```

- now that we have the variable ld1 which is a combination of height and weight, we can apply a multiple comparisions procedure to this new variable
- since variances are equal and distributions normal, the Tukey procedure is a reasonable choice
- we'll write the statistical conclusions on the next page

Posthoc Conclusions

At the 5% significance level there is strong evidence that the population mean vectors of height and weight for the apes at locations A and B both differ from those at location C. There is not a significant difference in population mean height and weight for the apes at locations A and B.

Posthoc Conclusions

At the 5% significance level there is strong evidence that the population mean vectors of height and weight for the apes at locations A and B both differ from those at location C. There is not a significant difference in population mean height and weight for the apes at locations A and B.

- no audio