Linear models for Qualitative Responses

Marco Antonio Florenzano Mollinetti[1]
[1]**University of Tsukuba, Systems Optimization Laboratory**
mollinetti@syou.cs.tsukuba.ac.jp

Before we Begin

- Go to the github repo:
  - https://github.com/Mollinetti/Experiment-Design-R
  - Download the script for this class! (in the ‘scripts’ folder, class_7.r!)
- Run the snippet at the beginning to load/install the required libraries

Agenda

- Introduction
- Classification of Quantitative responses
- Logistic Regression
- Linear Determinant Analysis (LDA)
- Quadratic Determinant Analysis (QDA)

- Introduction
- Classification of Quantitative responses
- Logistic Regression
- Linear Determinant Analysis (LDA)
- Quadratic Determinant Analysis (QDA)

Introduction

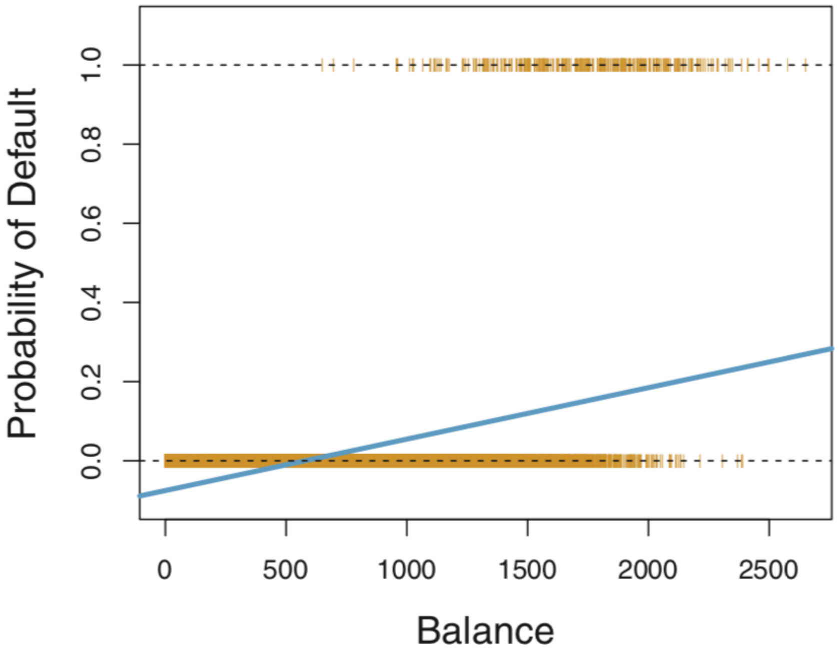
- Classification problems are frequent:
  1. Someone has a set of symptoms that can be attributed to three conditions. Which of the three does the person have?
  2. On a DNA sequence data of several patients with and without a disease, which DNA mutations are disease causing and which are not?

Introduction

- We will now fit a linear model to classify categorical variables
- For a 2 level qualitative response:
  - Logistic Regression*
- For more than two levels:
  - LDA
  - QDA

Introduction

- Why we can’t use linear regression anymore?



Answer: Negative Prediction values!

- Introduction
- Classification of Quantitative responses
- Logistic Regression
- Linear Determinant Analysis (LDA)
- Quadratic Determinant Analysis (QDA)

Classification of Quantitative Responses

- Remember, the response now has many levels
- For such, we can do a similar approach to the dummy variables
- Define your contrasts beforehand

$$Y = \begin{cases} 1 \text{ if response } 1 \\ 2 \text{ if response } 2 \\ 3 \text{ if response } 3 \end{cases}$$

or

$$Y = \begin{cases} 01 \text{ if response } 1 \\ 10 \text{ if response } 2 \\ 00 \text{ if response } 3 \end{cases}$$

Classification of Quantitative Responses

- How does a classifier outputs its predictions?
- Suppose we want to classify apples, oranges and pears
- We know by default that each variable is modeled in this specific way:

$$\text{apple} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{orange} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{pear} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Classification of Quantitative Responses

- Values outputted by the classifier close to the labels corresponds to each level:

$$\begin{bmatrix} 0.0002 \\ 0.9986 \end{bmatrix} \approx \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \text{apple}$$

$$\begin{bmatrix} 0.9780 \\ 0.0642 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \text{orange}$$

$$\begin{bmatrix} 0.9343 \\ 0.9846 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \text{pear}$$

- Introduction
- Classification of Quantitative responses
- Logistic Regression
- Linear Determinant Analysis (LDA)
- Quadratic Determinant Analysis (QDA)

Logistic Regression

- Logistic regression models that the probability that the response belongs to a category
- $p(X) = \Pr(Y = 1|X)$
- Best suited for two classes:
  - □ $p(X) < 0.5$ so Y belong to category 1
  - □ $p(X) \geq 0.5$ so Y belong to category 2

Logistic Regression

- If we follow the linear model (with additive and linear relations):

$$p(X) = \beta_0 + \beta_1 X_1 + \cdots \beta_p X_p + \epsilon$$

- Then $p(X)$ will have negative values, unacceptable for binary classification

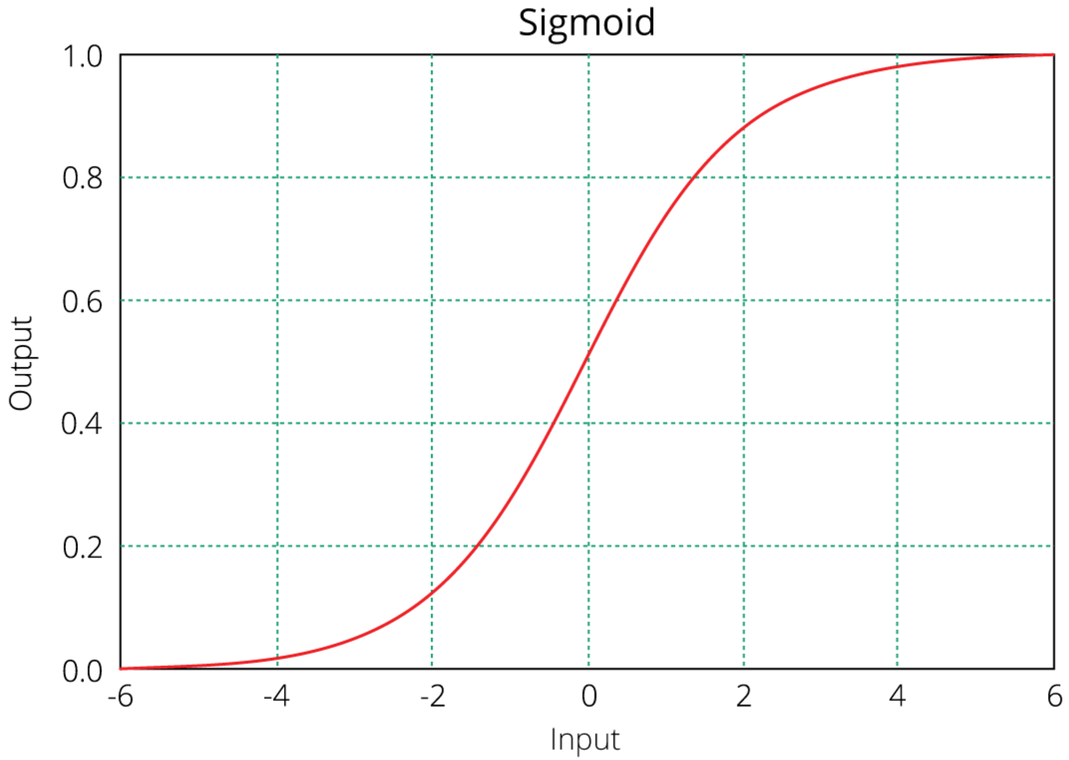
Logistic Regression

- $p(X)$ has to be a model in the $[0,1]$ interval
- This can be done by the sigmoid function

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p}}{1 + e^{\beta_0+\beta_1 X_1+\cdots+\beta_p}}$$

Logistic Regression

- The sigmoid function has the following shape:



Logistic Regression

- The model is now fitted using the maximum likelihood estimator
- The log-odds or logit captures the linear relation of the model:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}$$

- In a logistic regression, increasing X changes the log odds of the coefficients

Logistic Regression

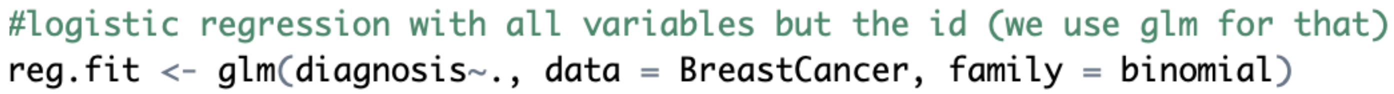
- Load the “BreastCancer” dataset from the mlbench library
- 33 columns (we’ll use 11)
- 1 qualitative variable (response)
  - 2 levels {benign, malignant}
- 10 quantitative variables (predictors)

Logistic Regression

- We will do first the regression with every quantitative variable, then we will adjust the model
- Verification of the assumptions are also required
- Goodness of fit:
  - Null Deviance (only the intercept)
  - Residual Deviance (all coefficients)
  - BIC/ AIC

Logistic Regression

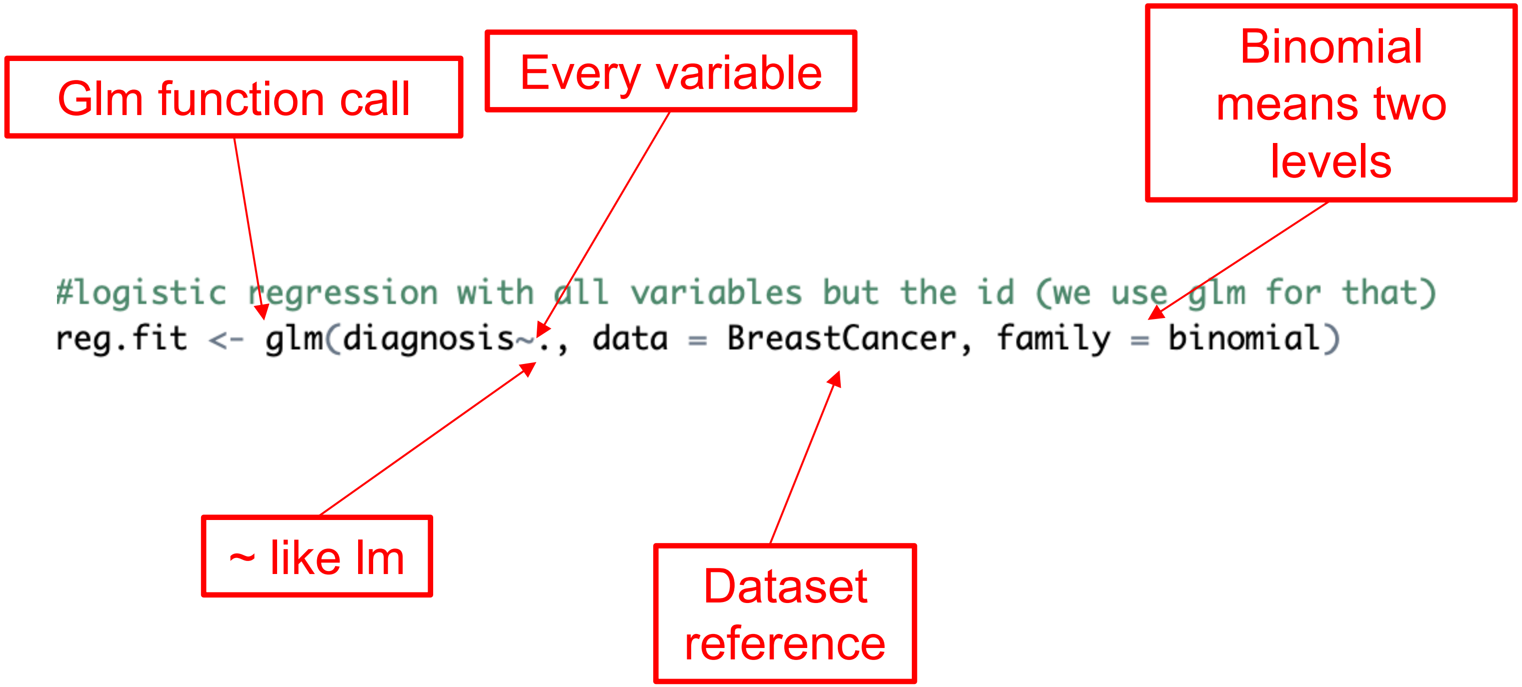
- In R, the glm is called for logistic regression



```
#logistic regression with all variables but the id (we use glm for that)
reg.fit <- glm(diagnosis~., data = BreastCancer, family = binomial)
```

Logistic Regression

- In R, the glm is called for logistic regression



```
#logistic regression with all variables but the id (we use glm for that)
reg.fit <- glm(diagnosis~., data = BreastCancer, family = binomial)
```

Glm function call

Every variable

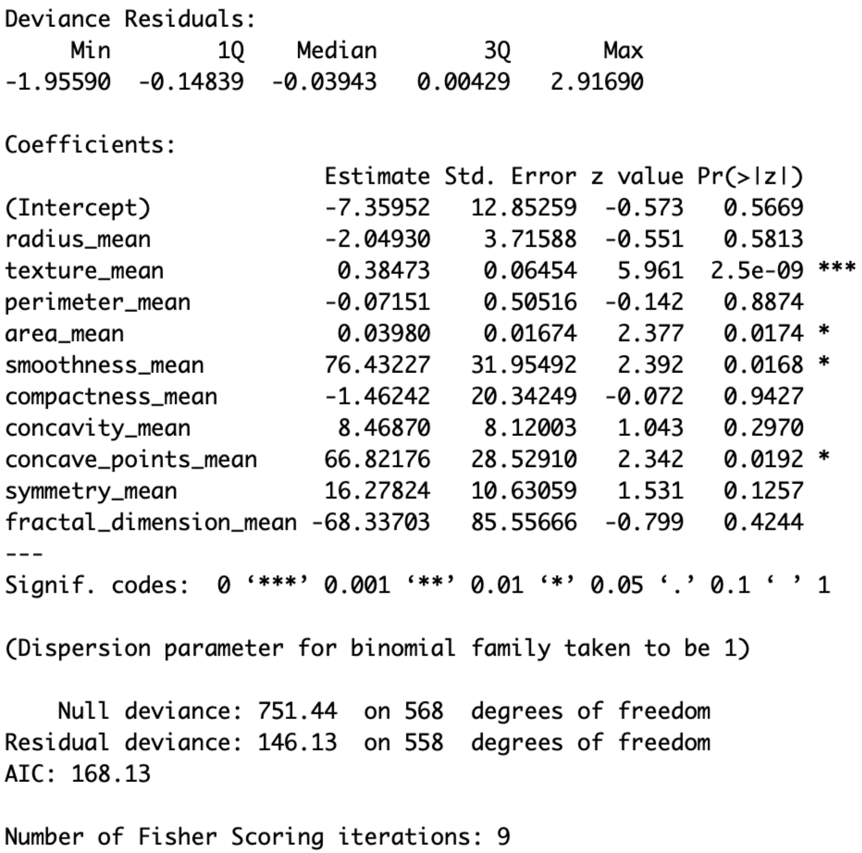
Binomial means two levels

~ like lm

Dataset reference

Logistic Regression

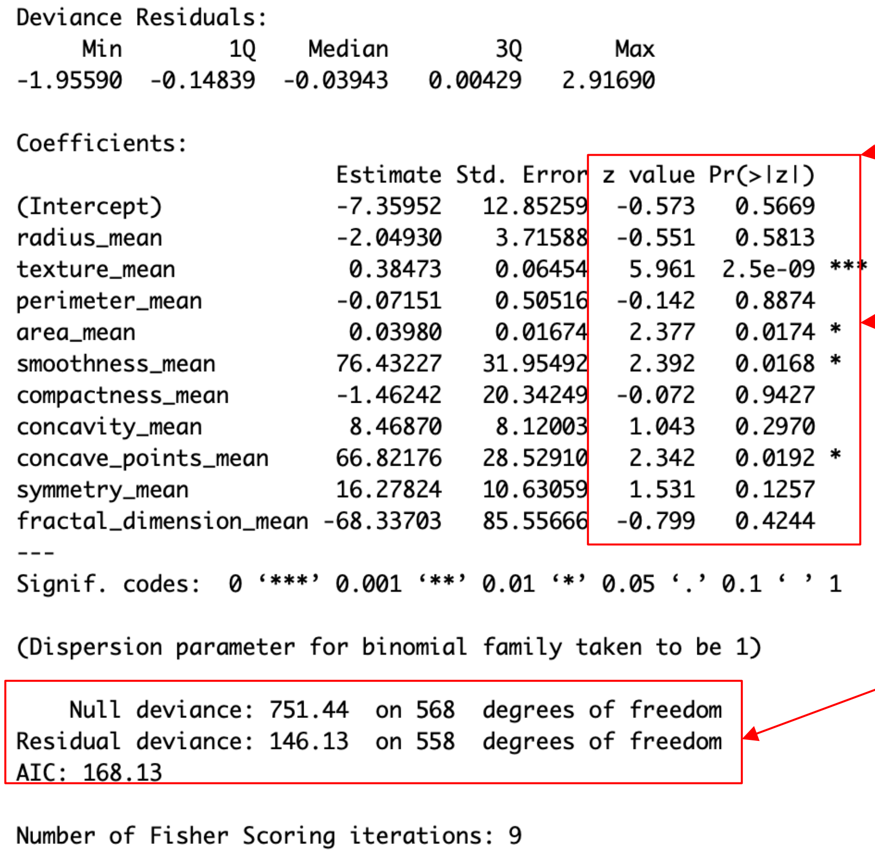
- Calling summary to the model gives the following info:



```
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.95590  -0.14839  -0.03943   0.00429   2.91690

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)             -7.35952   12.85259  -0.573   0.5669
radius_mean             -2.04930    3.71588  -0.551   0.5813
texture_mean             0.38473    0.06454   5.961  2.5e-09 ***
perimeter_mean          -0.07151    0.50516  -0.142   0.8874
area_mean                0.03980    0.01674   2.377   0.0174 *
smoothness_mean         76.43227   31.95492   2.392   0.0168 *
compactness_mean        -1.46242   20.34249  -0.072   0.9427
concavity_mean           8.46870    8.12003   1.043   0.2970
concave_points_mean     66.82176   28.52910   2.342   0.0192 *
symmetry_mean           16.27824   10.63059   1.531   0.1257
fractal_dimension_mean -68.33703   85.55666  -0.799   0.4244
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 751.44  on 568  degrees of freedom
Residual deviance: 146.13  on 558  degrees of freedom
AIC: 168.13

Number of Fisher Scoring iterations: 9
```

Logistic Regression

- Calling summary to the model gives the following info:



```
Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.95590  -0.14839  -0.03943   0.00429   2.91690

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)             -7.35952   12.85259  -0.573   0.5669
radius_mean             -2.04930    3.71588  -0.551   0.5813
texture_mean             0.38473    0.06454   5.961  2.5e-09 ***
perimeter_mean          -0.07151    0.50516  -0.142   0.8874
area_mean                0.03980    0.01674   2.377   0.0174 *
smoothness_mean         76.43227   31.95492   2.392   0.0168 *
compactness_mean        -1.46242   20.34249  -0.072   0.9427
concavity_mean           8.46870    8.12003   1.043   0.2970
concave_points_mean     66.82176   28.52910   2.342   0.0192 *
symmetry_mean           16.27824   10.63059   1.531   0.1257
fractal_dimension_mean -68.33703   85.55666  -0.799   0.4244
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 751.44  on 568  degrees of freedom
Residual deviance: 146.13  on 558  degrees of freedom
AIC: 168.13

Number of Fisher Scoring iterations: 9
```

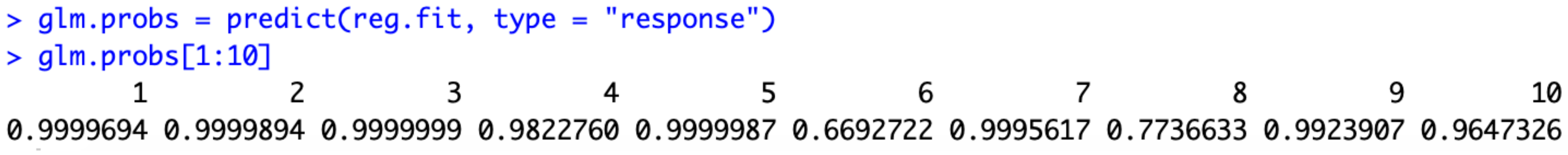
Z statistic:
Coefficient/standard error

Z statistic associated p-value

Goodness of Fit for
Logistic regression

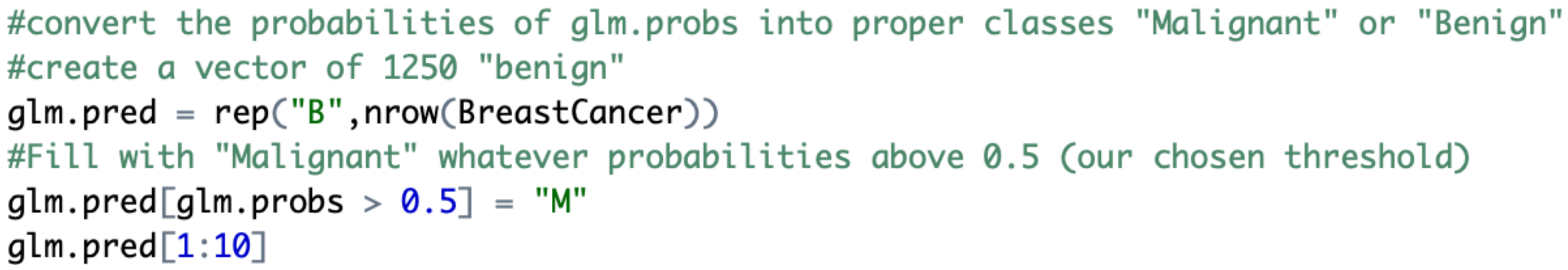
Logistic Regression

- We now predict the values
- Get the estimated value of each observation, then associate it to the class that it belongs

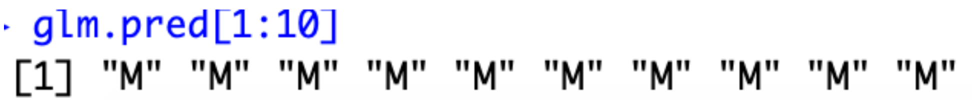


```
> glm.probs = predict(reg.fit, type = "response")
> glm.probs[1:10]
        1         2         3         4         5         6         7         8         9        10 
0.9999694 0.9999894 0.9999999 0.9822760 0.9999987 0.6692722 0.9995617 0.7736633 0.9923907 0.9647326 
```

Logistic Regression



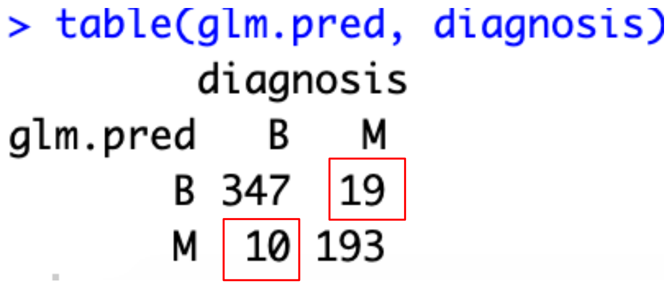
```
#convert the probabilities of glm.probs into proper classes "Malignant" or "Benign"
#create a vector of 1250 "benign"
glm.pred = rep("B",nrow(BreastCancer))
#Fill with "Malignant" whatever probabilities above 0.5 (our chosen threshold)
glm.pred[glm.probs > 0.5] = "M"
glm.pred[1:10]
```



```
> glm.pred[1:10]
 [1] "M" "M" "M" "M" "M" "M" "M" "M" "M" "M"
```

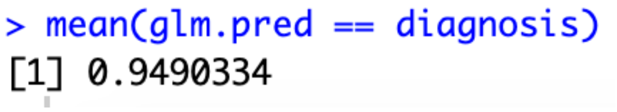
Logistic Regression

- Calculate a confusion matrix to check for false positive and false negatives



```
> table(glm.pred, diagnosis)
        diagnosis
glm.pred   B   M
       B 347  19
       M  10 193
```

- Check the mean of the predictions for accuracy



```
> mean(glm.pred == diagnosis)
[1] 0.9490334
```

ALWAYS REMEMBER TO CHECK BOTH

Logistic Regression

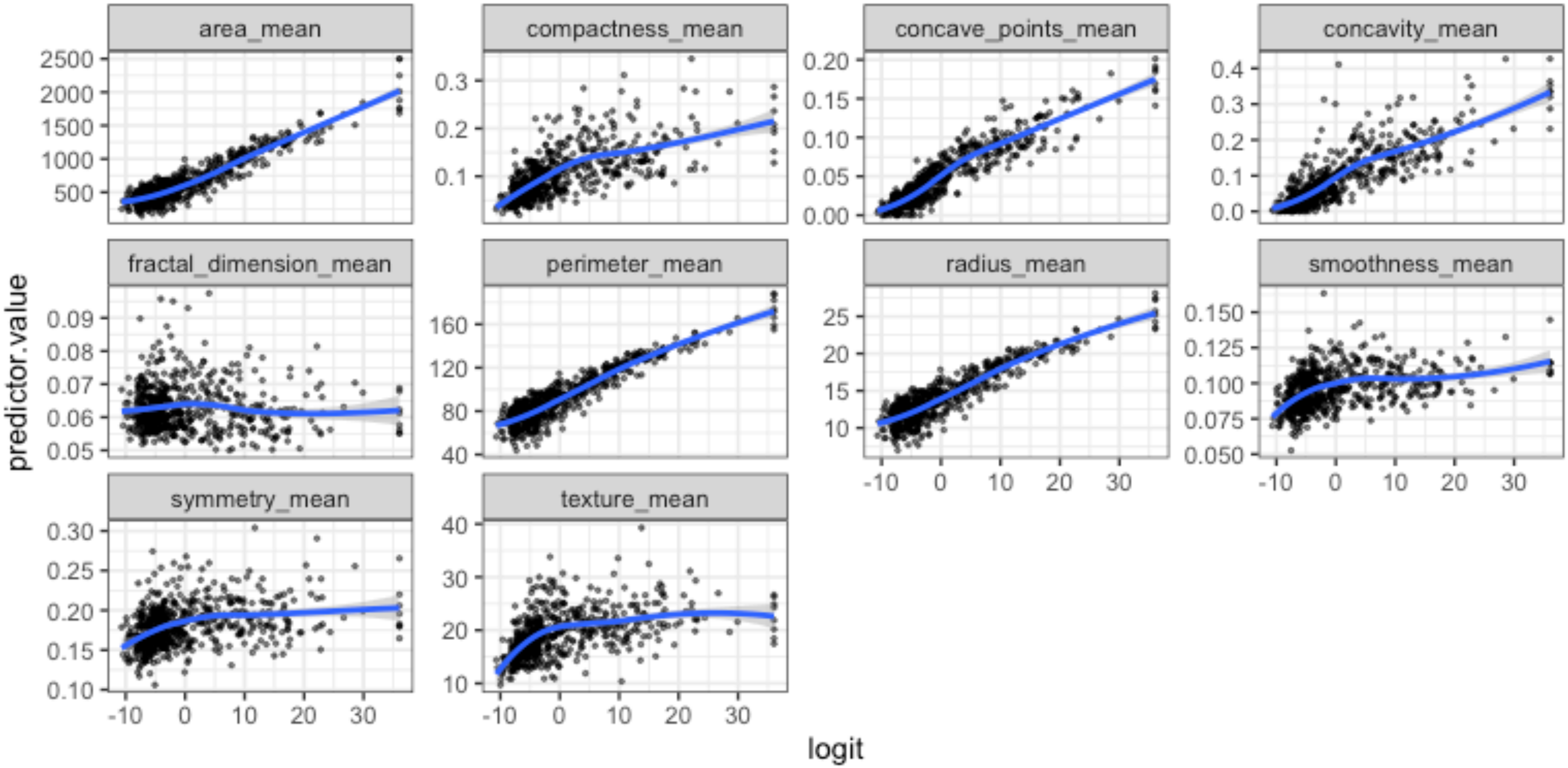
- Logistic regression does not require:
  - □ Normality
  - □ Homoscedascity
  - □ Linearity

Logistic Regression

- However, Logistic regression adhere to:
  - Linearity of independent variables and log odds
  - Influential values
  - Collinearity
  - Large sample size

Logistic Regression

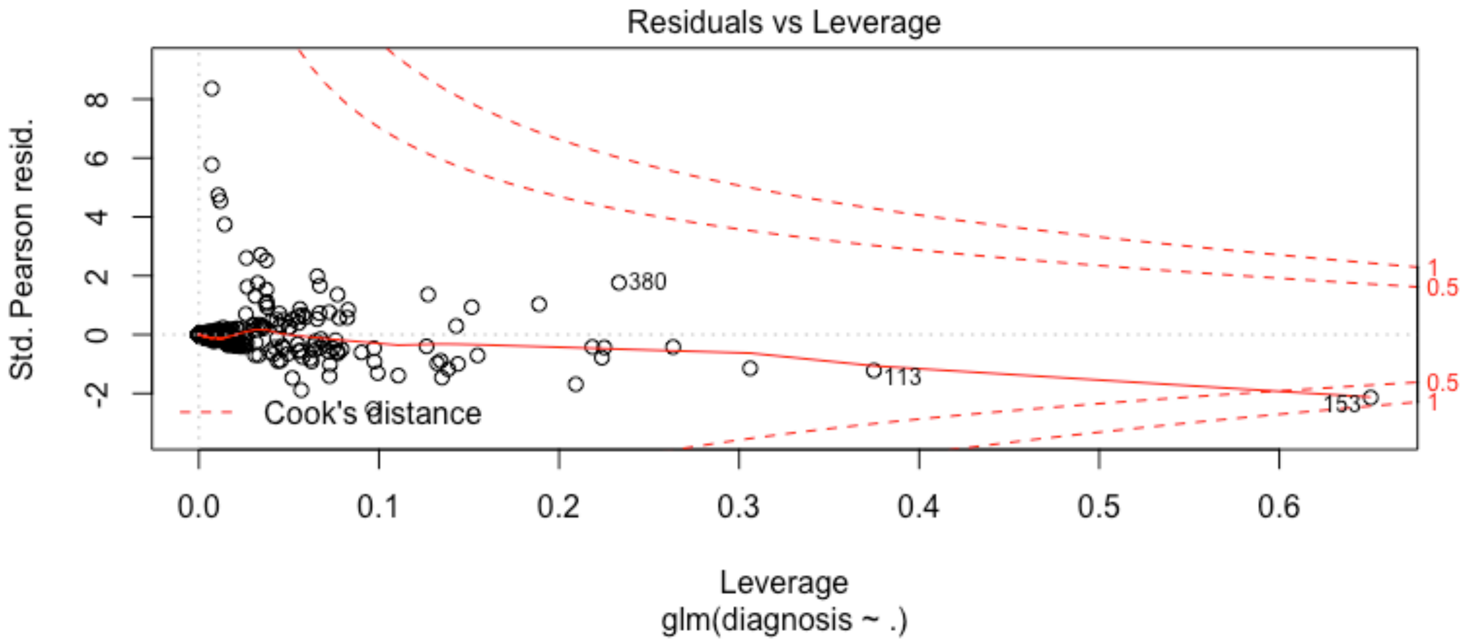
- Linearity of log odds
- inspecting the scatter plot between each predictor and the logit values


area_mean
compactness_mean
concave_points_mean
concavity_mean
fractal_dimension_mean
perimeter_mean
radius_mean
smoothness_mean
symmetry_mean
texture_mean
2500
2000
1500
1000
500
0.3
0.2
0.1
0.20
0.15
0.10
0.05
0.00
0.4
0.3
0.2
0.1
0.0
0.09
0.08
0.07
0.06
0.05
160
120
80
40
25
20
15
10
0.150
0.125
0.100
0.075
0.050
0.30
0.25
0.20
0.15
0.10
40
30
20
10
-10
0
10
20
30
predictor.value
logit

Logistic Regression

Influential Values

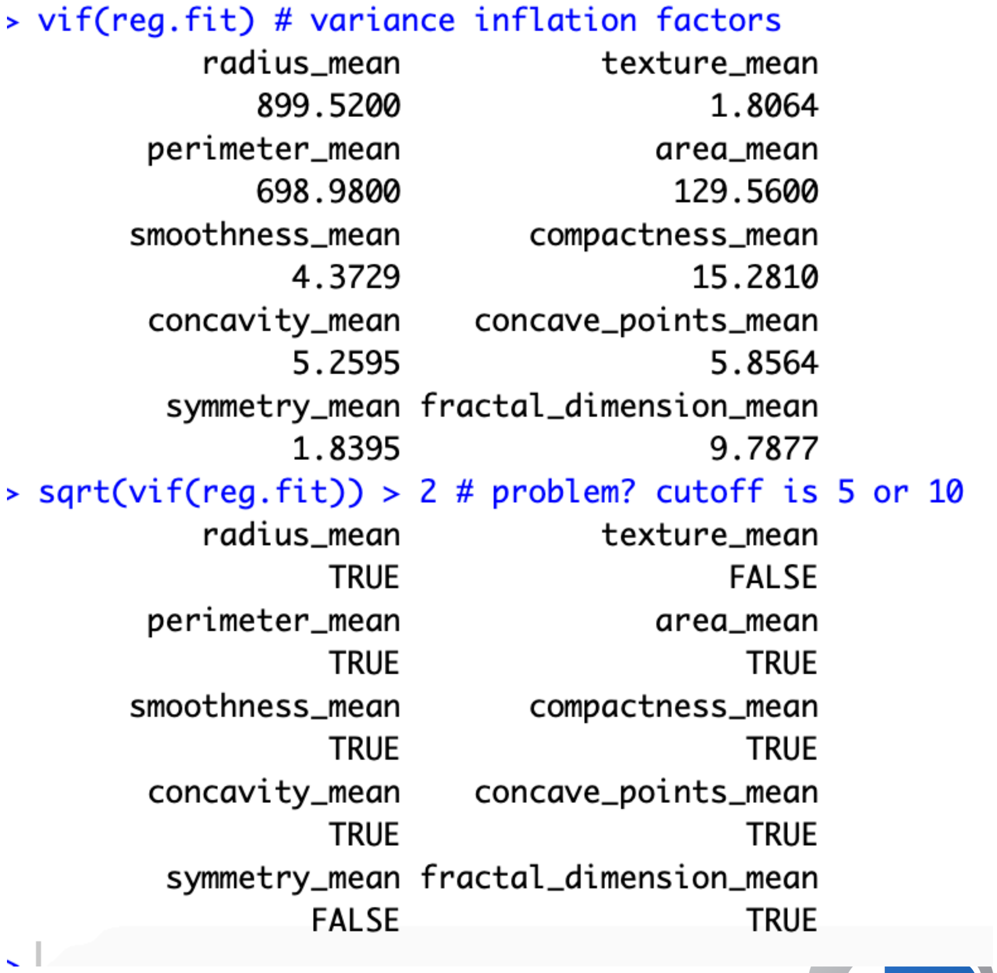
- □ Plot of the studentized residuals against leverage



Logistic Regression

Collinearity

- Check the correlation
- Compute the VIF



```
> vif(reg.fit) # variance inflation factors
           radius_mean            texture_mean 
              899.5200                  1.8064 
        perimeter_mean               area_mean 
              698.9800                129.5600 
       smoothness_mean        compactness_mean 
                4.3729                 15.2810 
        concavity_mean     concave_points_mean 
                5.2595                  5.8564 
         symmetry_mean fractal_dimension_mean 
                1.8395                  9.7877 
> sqrt(vif(reg.fit)) > 2 # problem? cutoff is 5 or 10
           radius_mean            texture_mean 
                  TRUE                   FALSE 
        perimeter_mean               area_mean 
                  TRUE                    TRUE 
       smoothness_mean        compactness_mean 
                  TRUE                    TRUE 
        concavity_mean     concave_points_mean 
                  TRUE                    TRUE 
         symmetry_mean fractal_dimension_mean 
                 FALSE                    TRUE 
> 
```

- Introduction
- Classification of Quantitative responses
- Logistic Regression
- Linear Determinant Analysis (LDA)
- Quadratic Determinant Analysis (QDA)

Linear Determinant Analysis (LDA)

- Logistic Regression is not recommended for more than 2 classes
- Does not work well when classes are well separated
- LDA is more stable for smaller and normal X
- Assumes that probabilities are distributed from a gaussian distribution

Linear Determinant Analysis (LDA)

- Let Y be able to take on distinct K values of classes
- LDA models the probability of X belonging to a Y as an approximation of the Bayes' Theorem

$$\Pr(Y = K | X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

Linear Determinant Analysis (LDA)

- Let Y be able to take on distinct K values of classes
- LDA models the probability of X belonging to a Y as an approximation of the Bayes' Theorem

Prior probability

$$\Pr(Y = K|X = x) = \frac{\boxed{\pi_k} f_k(x)}{\sum_{l=1}^{K} \pi_l \boxed{f_l}(x)}$$

Posterior probability (Density function)

Linear Determinant Analysis (LDA)

- For a multivariate gaussian distribution the density function is:

Variance

$$f(x) = \frac{1}{(\boxed{2\pi})^{p/2}\boxed{|\Sigma|}^{1/2}} \exp\left(-\frac{1}{2}(x - \boxed{\mu})^T \Sigma^{-1}(x - \mu)\right).$$

Covariance matrix

Mean

Linear Determinant Analysis (LDA)

- Values of parameters $\mu_1, \ldots, \mu_K$ and $\pi_1, \ldots, \pi_K$must be estimated for each class
- Each value is calculated as:

$$\hat{\mu}_K = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2{}_K = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} x_i$$

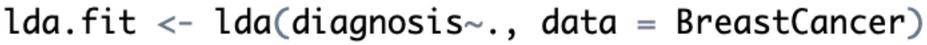
Linear Determinant Analysis (LDA)

- The estimated values are plugged into the following formula and classified as the one that maximizes for given class
- The covariance matrix is the same for each class

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

Linear Determinant Analysis (LDA)

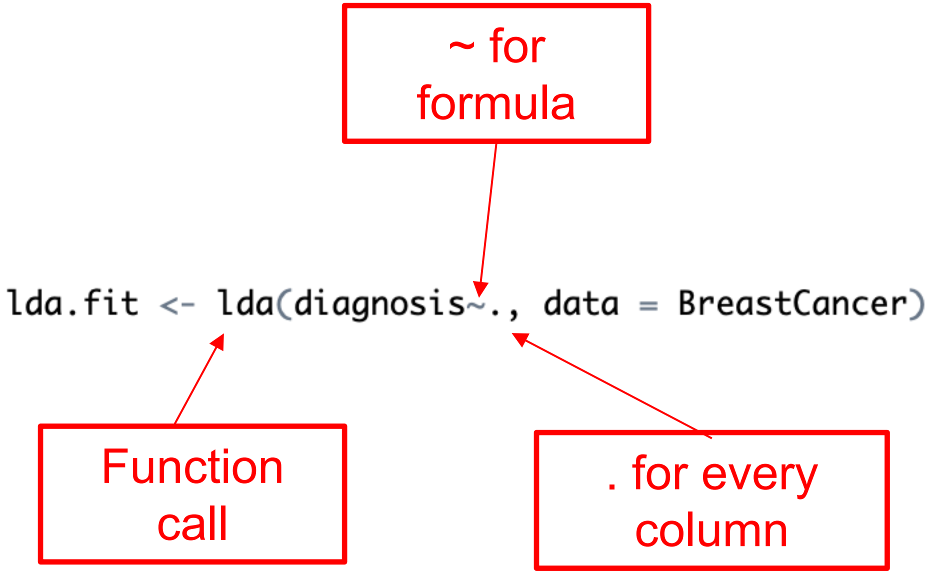
- In R, LDA is part of the MASS library



```
lda.fit <- lda(diagnosis~., data = BreastCancer)
```

Linear Determinant Analysis (LDA)

- In R, LDA is part of the MASS library



~ for formula

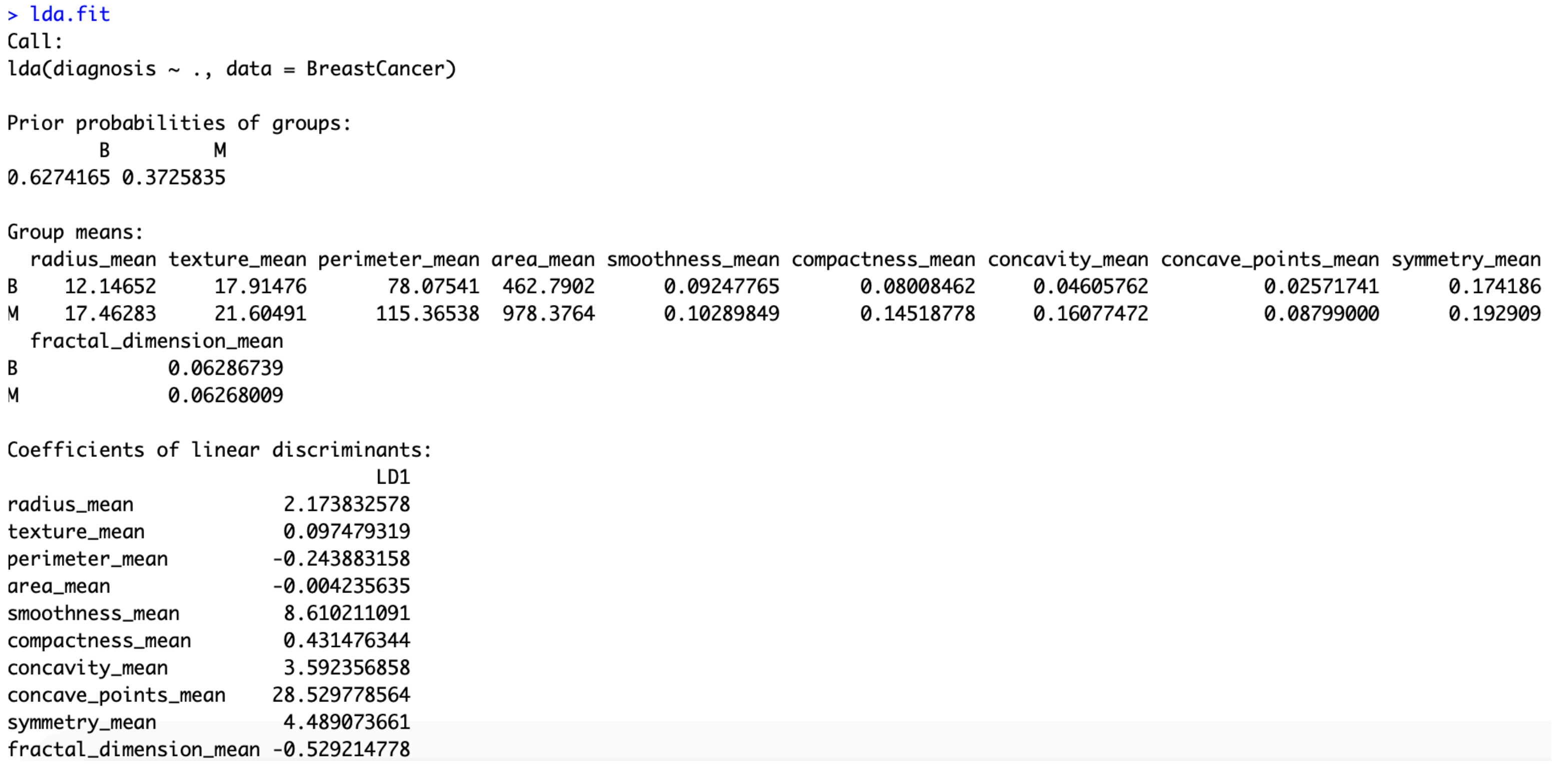
```
lda.fit <- lda(diagnosis~., data = BreastCancer)
```

Function call

. for every column

Linear Determinant Analysis (LDA)

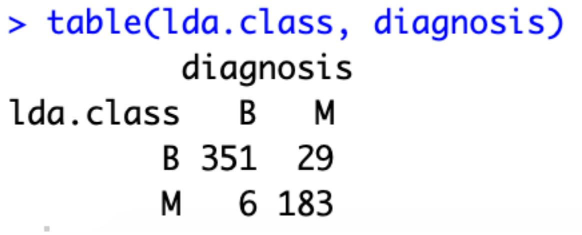
- Calling the object:



```
> lda.fit
Call:
lda(diagnosis ~ ., data = BreastCancer)

Prior probabilities of groups:
        B         M 
0.6274165 0.3725835 

Group means:
  radius_mean texture_mean perimeter_mean area_mean smoothness_mean compactness_mean concavity_mean concave_points_mean symmetry_mean
B    12.14652     17.91476       78.07541  462.7902      0.09247765       0.08008462     0.04605762          0.02571741      0.174186
M    17.46283     21.60491      115.36538  978.3764      0.10289849       0.14518778     0.16077472          0.08799000      0.192909
  fractal_dimension_mean
B             0.06286739
M             0.06268009

Coefficients of linear discriminants:
                                LD1
radius_mean              2.173832578
texture_mean             0.097479319
perimeter_mean          -0.243883158
area_mean               -0.004235635
smoothness_mean          8.610211091
compactness_mean         0.431476344
concavity_mean           3.592356858
concave_points_mean     28.529778564
symmetry_mean            4.489073661
fractal_dimension_mean  -0.529214778
```

Logistic Regression

- Calculate yet again a confusion matrix to check for false positives and false negatives



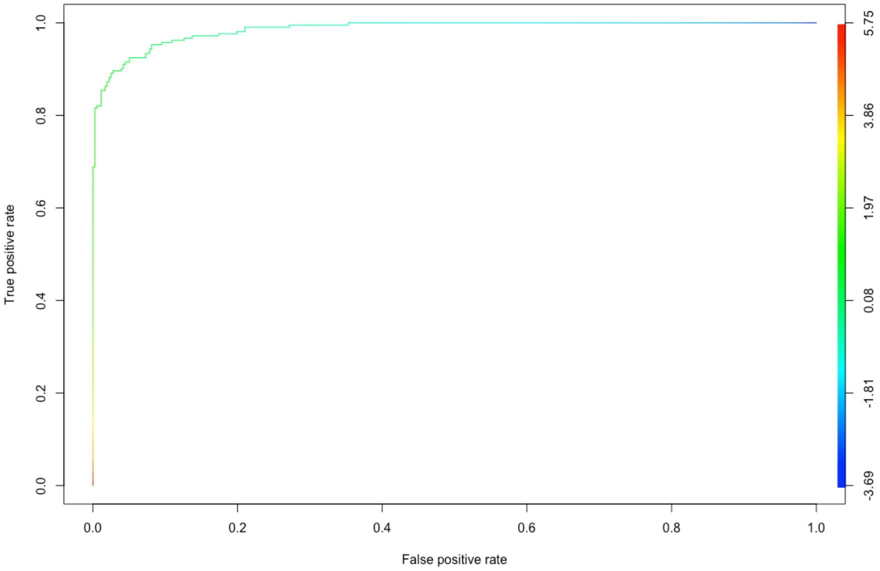
```
> table(lda.class, diagnosis)
         diagnosis
lda.class   B   M
        B 351  29
        M   6 183
```

Linear Determinant Analysis (LDA)

- Class-specific accuracy is more important than general accuracy in some cases
- Possible way to improve: change the threshold values
- However, how can we decide the best threshold value?

Linear Determinant Analysis (LDA)

- Plot of the ROC curve
- Plot all possible thresholds and measure the AUC (area under curve)
- A curve that hugs the top left is desired



Linear Determinant Analysis (LDA)

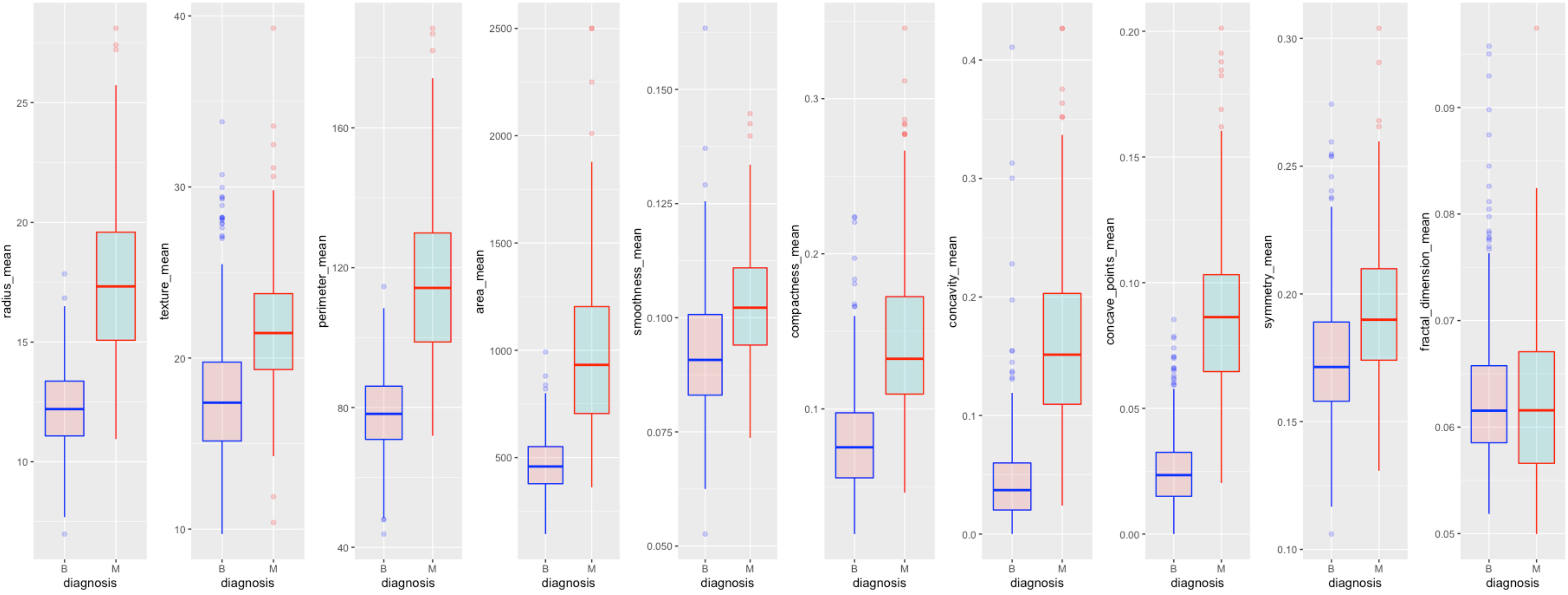
- Assumptions:
  - Equality of Variance-Covariance
  - Normality

Linear Determinant Analysis (LDA)

- Equality of Variance-Covariance:
  - Plot of the boxplots
  - Plot of the covariances homogeneity
  - Box M-test

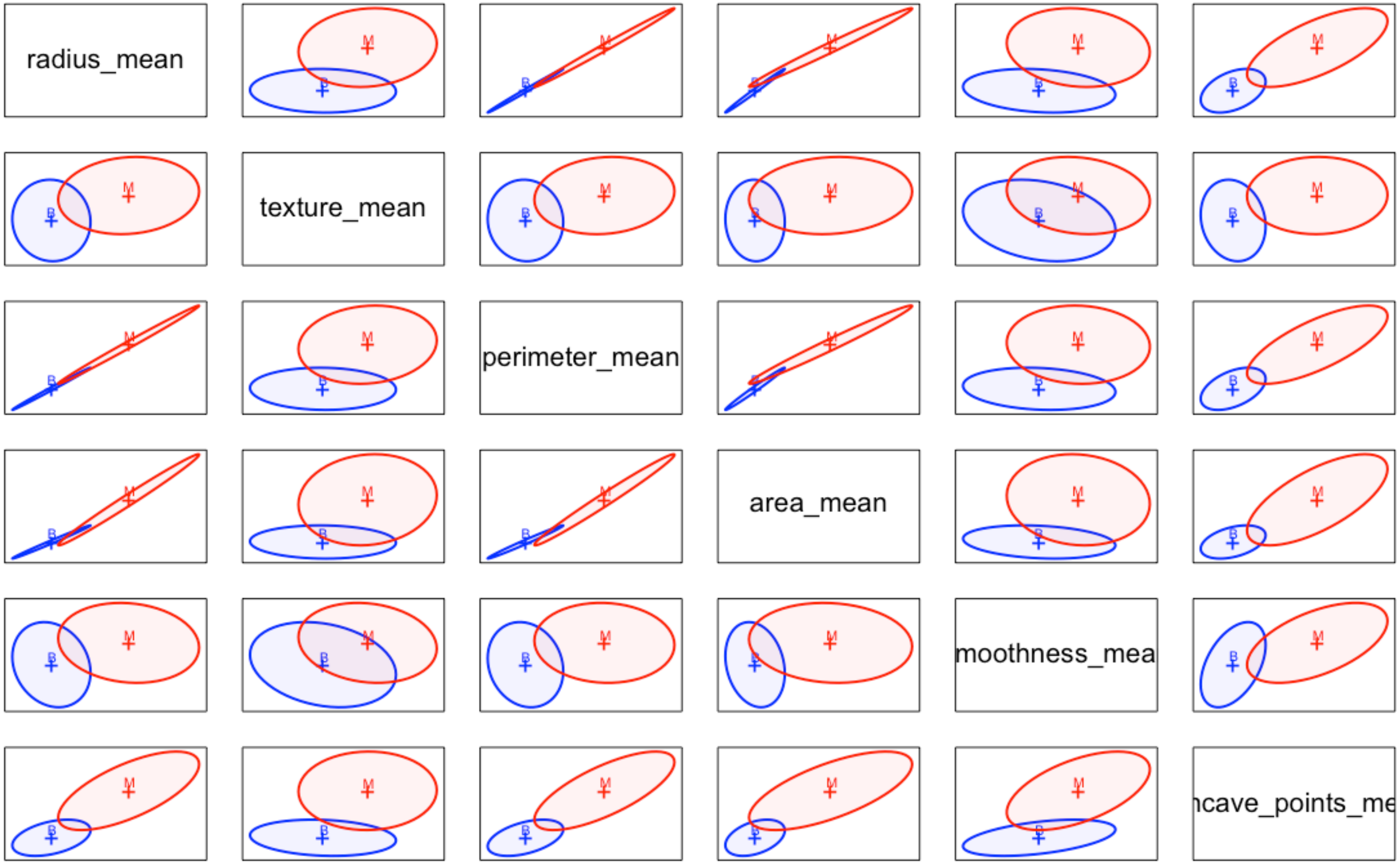
Linear Determinant Analysis (LDA)

- Plot the boxplots of variances between classes



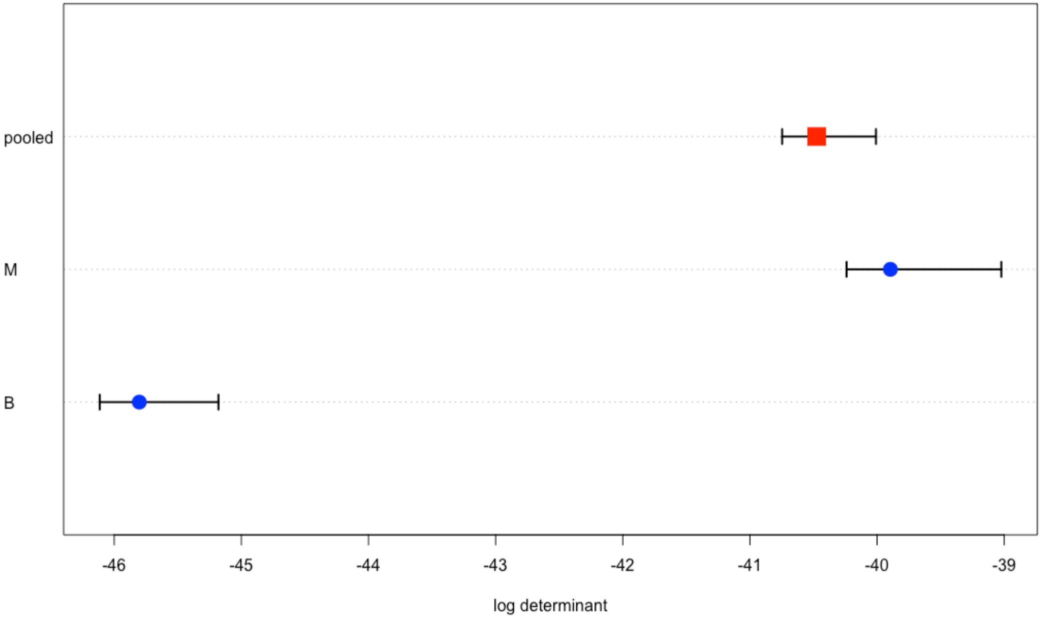
Linear Determinant Analysis (LDA)

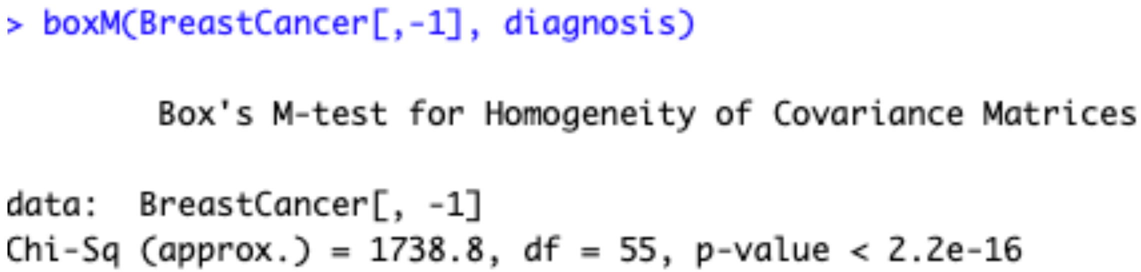
- Plot of the covariances homogeneity



Linear Determinant Analysis (LDA)

- Box M-test





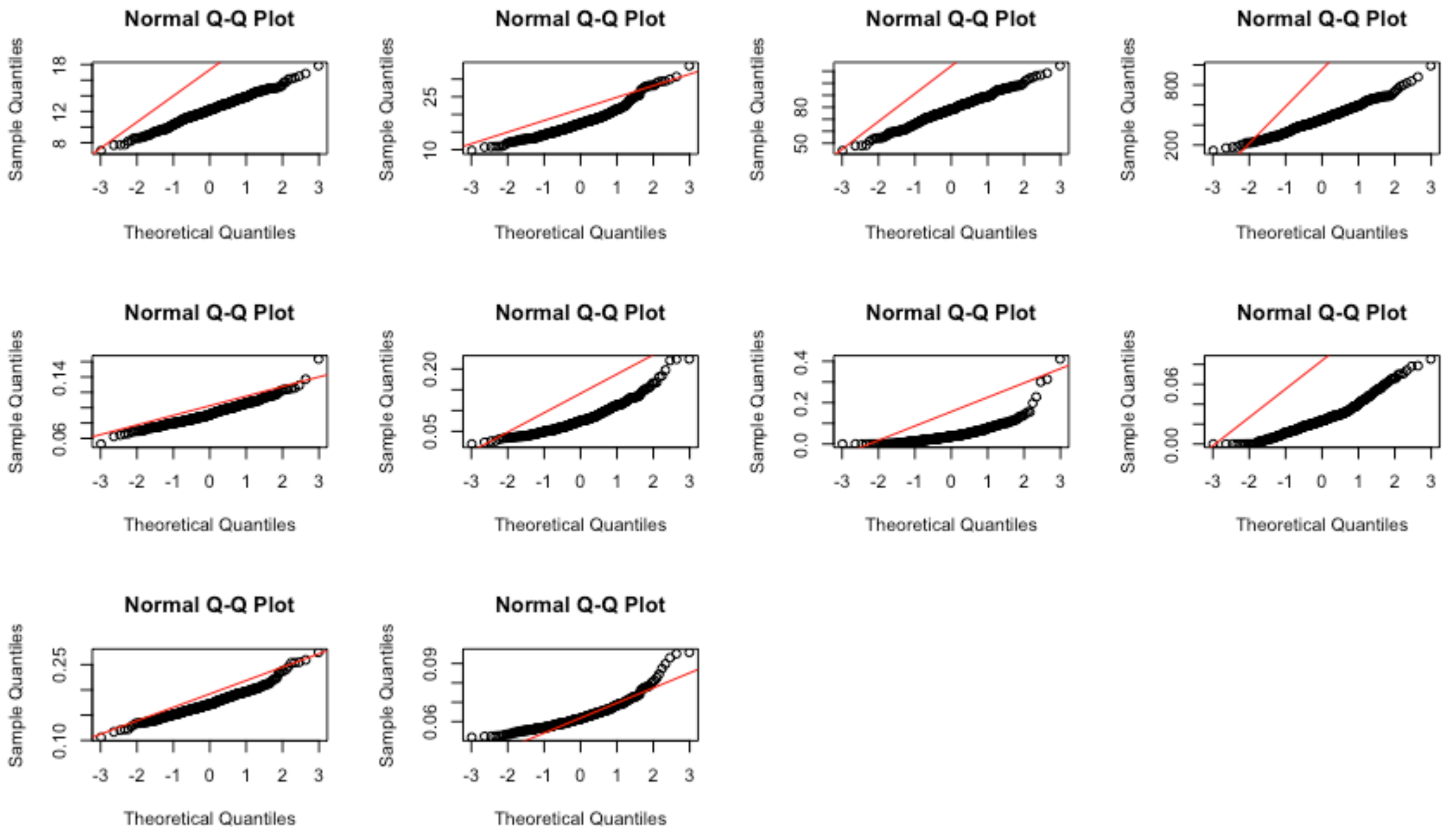
```
> boxM(BreastCancer[,-1], diagnosis)

	Box's M-test for Homogeneity of Covariance Matrices

data:  BreastCancer[, -1]
Chi-Sq (approx.) = 1738.8, df = 55, p-value < 2.2e-16
```

Linear Determinant Analysis (LDA)

- Normality:
  - QQ-plots
  - Shapiro-wilk test

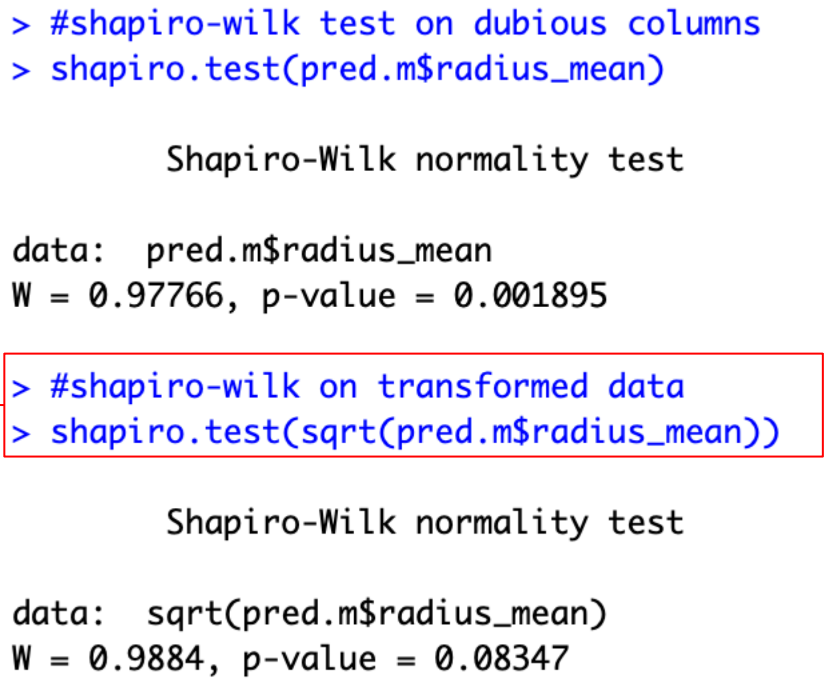
Linear Determinant Analysis (LDA)

- QQ-plots: plotting the predicted value against the predictor



Linear Determinant Analysis (LDA)

- For predictors that are clearly not normal, run the shapiro-wilk test against predicted values



```
> #shapiro-wilk test on dubious columns
> shapiro.test(pred.m$radius_mean)

	Shapiro-Wilk normality test

data:  pred.m$radius_mean
W = 0.97766, p-value = 0.001895

> #shapiro-wilk on transformed data
> shapiro.test(sqrt(pred.m$radius_mean))

	Shapiro-Wilk normality test

data:  sqrt(pred.m$radius_mean)
W = 0.9884, p-value = 0.08347
```

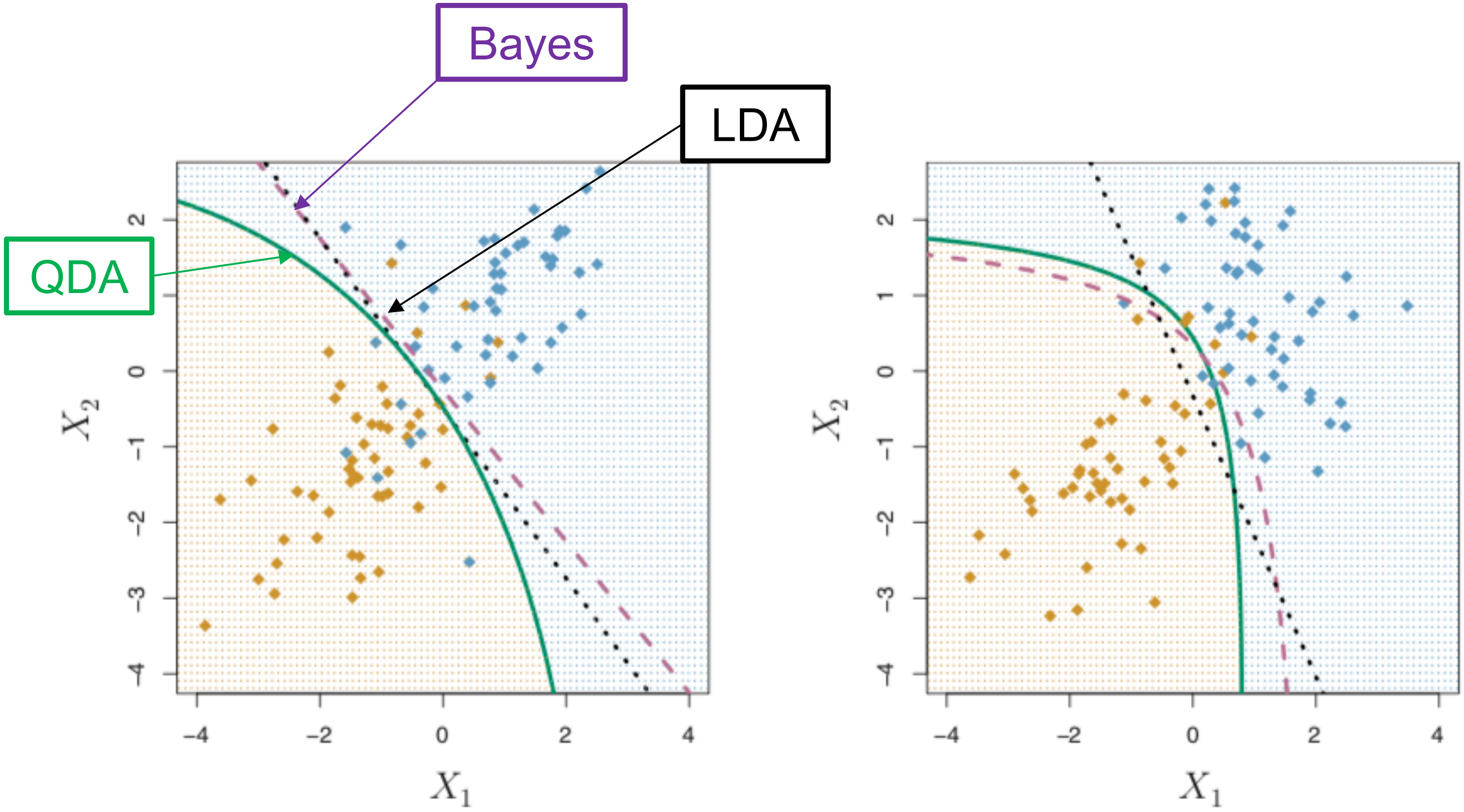
Transform the data if not normal

- Introduction
- Classification of Quantitative responses
- Logistic Regression
- Linear Determinant Analysis (LDA)
- Quadratic Determinant Analysis (QDA)

Quadratic Determinant Analysis

- QDA still assumes that probabilities are drawn from a multivariate gaussian distribution
- However, each class now has its own covariance matrix
- More flexible to fit curves

Quadratic Determinant Analysis



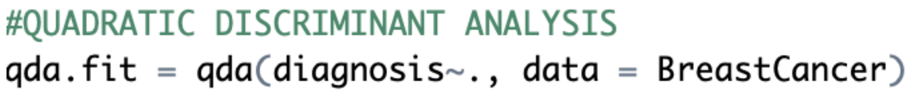
Quadratic Determinant Analysis

- Estimated values are plugged into the following function:

$$\delta_k(x) = -\frac{1}{2}x^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\log|\Sigma_k| + \log\pi_k$$

Quadratic Determinant Analysis

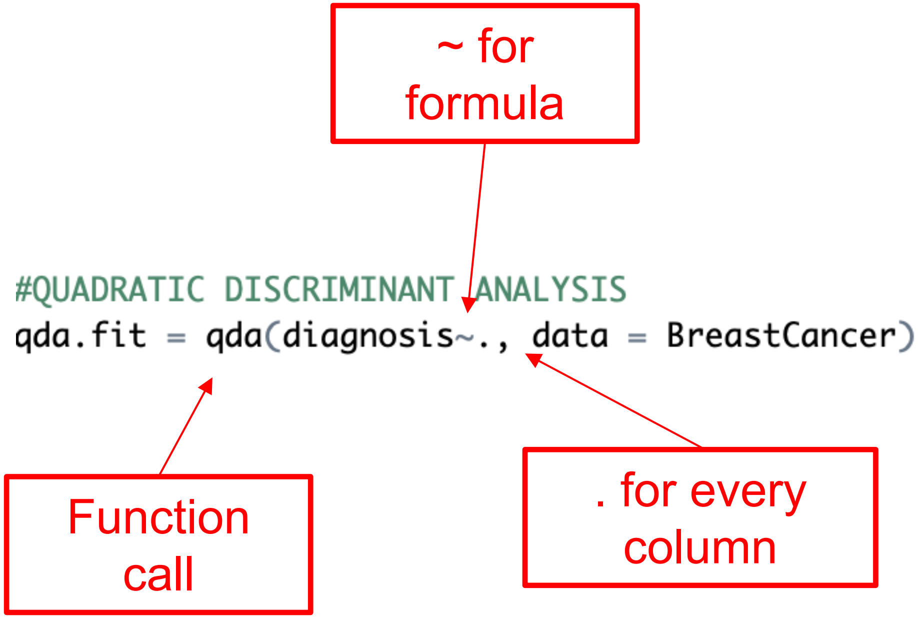
- In R, QDA is also part of the MASS library



```
#QUADRATIC DISCRIMINANT ANALYSIS
qda.fit = qda(diagnosis~., data = BreastCancer)
```

Quadratic Determinant Analysis

- In R, QDA is also part of the MASS library



```
#QUADRATIC DISCRIMINANT ANALYSIS
qda.fit = qda(diagnosis~., data = BreastCancer)
```

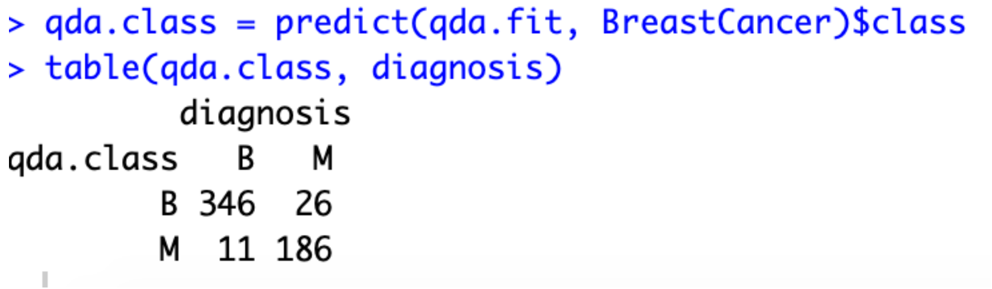
~ for formula

Function call

. for every column

Quadratic Determinant Analysis

- Predictions and confusion matrix must also be estimated



```
> qda.class = predict(qda.fit, BreastCancer)$class
> table(qda.class, diagnosis)
         diagnosis
qda.class   B   M
        B 346  26
        M  11 186
```

Quadratic Determinant Analysis

- Assumptions of QDA are the same as LDA
  - Equality of Variance-Covariance
  - Normality

Next Episode

- Now that we understand how to fit lines, we will go back to the Multiple Analysis of Variances (MANOVA)

Multiple Linear Regression Model

Marco Antonio Florenzano Mollinetti[1]
[1]**University of Tsukuba, Systems Optimization Laboratory**
mollinetti@syou.cs.tsukuba.ac.jp