# Day 3: Classification

Lucas Leemann

Essex Summer School

Introduction to Statistical Learning

1. Motivation for Classification
2. Logistic Regression
    - The Linear Probability Model
    - Building a Model from Probability Theory
3. Linear Discriminant Analysis
    - Building a Model from Probability Theory
    - Example 1 (k=2)
    - Example 2
4. Comparison of Classification Methods

# Classification

Standard data science problem, i.e.

- who will default on credit loan?
- which customers will come back?
- which e-mails are spam?
- which ballot stations manipulated the vote returns?
- who is likely to vote for which party?

# Logistic Regression

# Linear Probability Model

## LPM

The linear probability model relies on linear regression to analyze binary variables.

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 \cdot X_{1i} + \beta_2 \cdot X_{2i} + ... + \beta_k \cdot X_{ki} + \varepsilon_i \\ Pr(Y_i = 1 | X_1, X_2, ...) &= \beta_0 + \beta_1 \cdot X_{1i} + \beta_2 \cdot X_{2i} + ... + \beta_k \cdot X_{ki} \end{aligned}$$

Advantages:

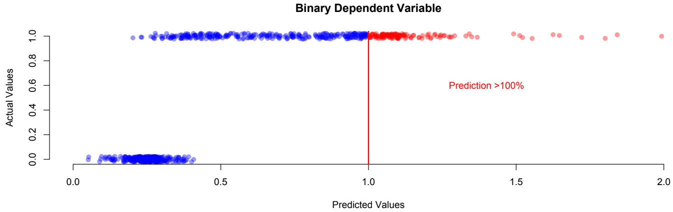
- We can use a well-known model for a new class of phenomena
- Easy to interpret the marginal effects of $X$

## Problems with Linear Probability Model

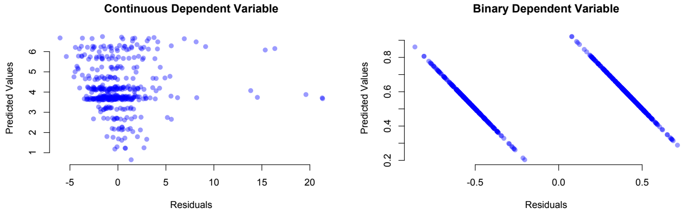
The linear model needs a continuous dependent variable, if the dependent variable is binary we run into problems:

- Predictions, $\hat{y}$, are interpreted as probability for $y = 1$
  $\rightarrow P(y = 1) = \hat{y} = \beta_0 + \beta_1 X$, can be above 1 if $X$ is large enough
  $\rightarrow P(y = 0) = \hat{y} = \beta_0 + \beta_1 X$, can be below 0 if $X$ is small enough
- The errors will not have a constant variance.
  $\rightarrow$ For a given $X$ the residual can be either $(1\text{-}\beta_0\text{-}\beta_1 X)$ or $(\beta_0 + \beta_1 X)$
- The linear function might be wrong
  $\rightarrow$ Imagine you buy a car. Having an additional 1000£ has a very different effect if you are broke or if you already have another 12,000£ for a car.

# Predictions can lay outside $I = [0, 1]$

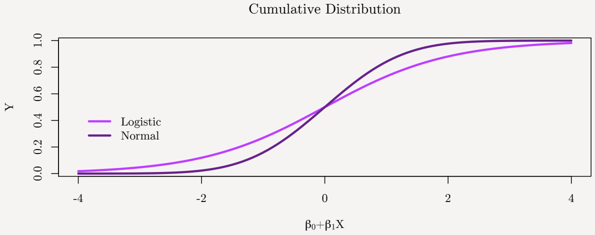



# Residuals if the dependent variable is binary:



## Predictions should only be within $I = [0, 1]$

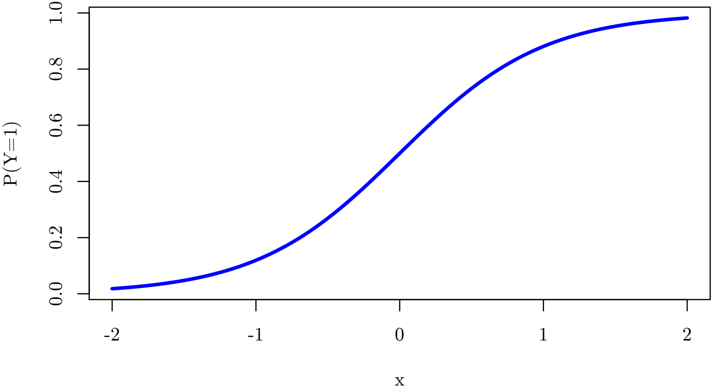
- We want to make predictions in terms of probability
- We can have a model like this: $P(y_i = 1) = F(\beta_0 + \beta_1 X_i)$
  where $F(\cdot)$ should be a function which never returns values below 0 or above 1
- There are two possibilities for $F(\cdot)$: cumulative normal ($\Phi$) or logistic ($\Delta$) distribution



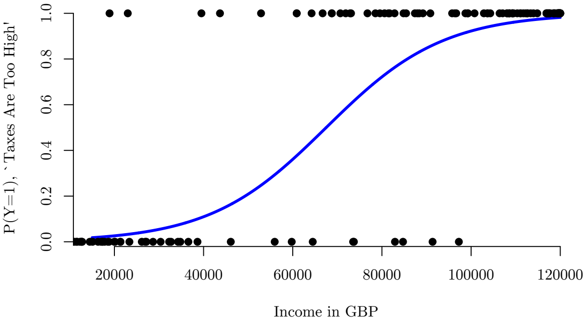
# Logit Model

The logit model is then: $P(y_i = 1) = \frac{1}{1+\exp(-\beta_0 - \beta_1 X_i)}$

For $\beta_0 = 0$ and $\beta_1 = 2$ we get:

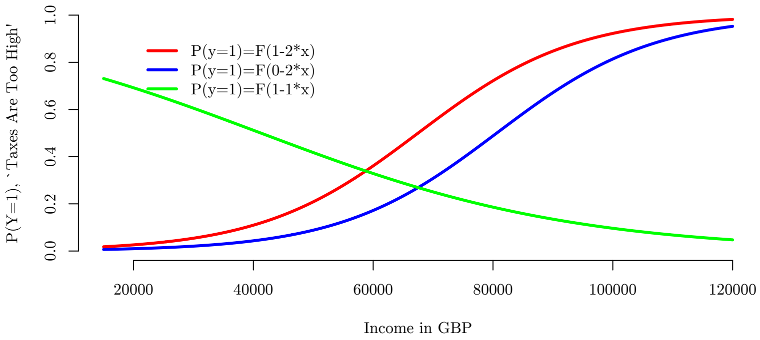


# Logit Model: Example




- We can make a prediction by calculating: $P(y=1) = \frac{1}{1+\exp(-\beta_0 - \beta_1 \cdot X)}$

# Logit Model: Example




- A positive $\beta_1$ makes the s-curve increase.
- A smaller $\beta_0$ shifts the s-curve to the right.
- A negative $\beta_1$ makes the s-curve decrease.

# Example: Women in the 1980s and Labour Market

```
> m1 <- glm(inlf ~ kids + age + educ, dat=data1, family=binomial(logit))
> summary(m1)

Call:
glm(formula = inlf ~ kids + educ + age, family = binomial(logit),
    data = data1)
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8731  -1.2325   0.8026   1.0564   1.5875

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.11437    0.73459  -0.156  0.87628
kids        -0.50349    0.19932  -2.526  0.01154 *
educ         0.16902    0.03505   4.822 1.42e-06 ***
age         -0.03108    0.01137  -2.734  0.00626 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1029.75  on 752  degrees of freedom
Residual deviance:  993.53  on 749  degrees of freedom
AIC: 1001.5
```

# Example: Women 1980 (2)

```
Call:
glm(formula = inlf ~ kids + educ + age, family = binomial(logit),
    data = data1)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.11437    0.73459  -0.156  0.87628
kids        -0.50349    0.19932  -2.526  0.01154 *
educ         0.16902    0.03505   4.822 1.42e-06 ***
age         -0.03108    0.01137  -2.734  0.00626 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

- Only interpret direction and significance of a coefficient
- The test statistic always follows a normal distribution ($z$)

# Example: Women 1980 (3)

```
glm(formula = inlf ~ kids + educ + age, family = binomial(logit),
    data = data1)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.11437    0.73459  -0.156  0.87628
kids        -0.50349    0.19932  -2.526  0.01154 *
educ         0.16902    0.03505   4.822 1.42e-06 ***
age         -0.03108    0.01137  -2.734  0.00626 **
```

- How can we generate a prediction for a woman with no kids, 13 years of education, who is 32?
    - Compute first the prediction on $y^*$, i.e. just compute $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$
    - $P(y=1) = \frac{1}{1+exp(0.11+.50\cdot 0-0.17\cdot 13+0.03\cdot 32)} = \frac{1}{1+exp(-1.09)} = 0.75$

# Prediction

```
> z.out1 <- zelig(inlf ~ kids + age + educ + exper + huseduc + huswage, model = "logit", data = data1)
```

```
> average.woman <- setx(z.out1, kids=median(data1$kids), age=mean(data1$age), educ=mean(data1$educ),
                        exper=mean(data1$exper), huseduc=mean(data1$huseduc), huswage=mean(data1$huswage))
> s.out <- sim(z.out1,x=average.woman)
> summary(s.out)
```

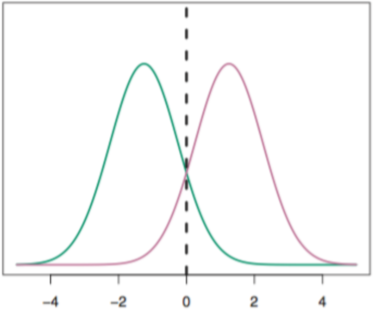
```
 sim x :
 -----
ev
          mean         sd       50%      2.5%     97.5%
[1,] 0.5746569 0.02574396 0.5754419 0.5232728 0.6217502
pv
         0     1
[1,] 0.432 0.568
```

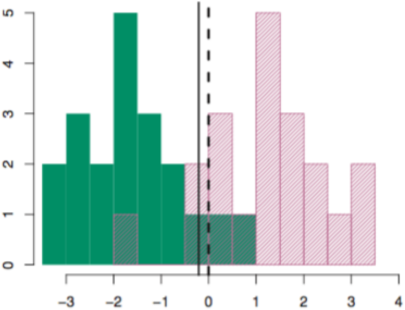
# Linear Discriminant Analysis

# Linear Discriminant Analysis

- Why something new?
  - Might have more than 3 classes
  - problems of separation
- Basic idea: We try to learn about $Y$ by looking at the distribution of $X$
- Logistic regression did this: $Pr(Y = k|X = x)$
- LDA will exploit Bayes' theorem and infer class probability directly from $X$ and prior probabilities

# Basic Idea: Linear Discriminant Analysis




(James et al, 2013: 140)

# Math-Stat Refresher: Bayes

Doping tests:

- 99% sensitive (correctly identifies doping abuse), $P(+|D) = .99$
- 99% specific (correctly identifies appropriate behavior), $P(-|noD) = .99$
- 0.5% athletes take illegal substances
- You take a test and receive a positive result. What is the probability that you actually took an illegal substance?

$$P(D|+) = \frac{P(D) \cdot P(+|D)}{P(D) \cdot P(+|D) + P(noD) \cdot P(+|noD)}$$

$$P(D|+) = \frac{0.005 \cdot 0.99}{0.005 \cdot 0.99 + 0.995 \cdot 0.01} = 0.332$$

# LDA: The Mechanics (with one $X$)

- We have $X$ and it follows a distribution $f(x)$
- We have $k$ different classes
- Based on $Y$, we can calculate the prior probabilities $\pi_k$

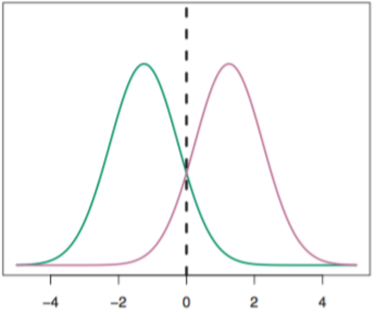
1. Define $f_k(x)$ as the distribution of $X$ for class $k$ (p. 140/141)
2. Note: $f_k(x) = P(X = x|Y = k)$
3. Hence:

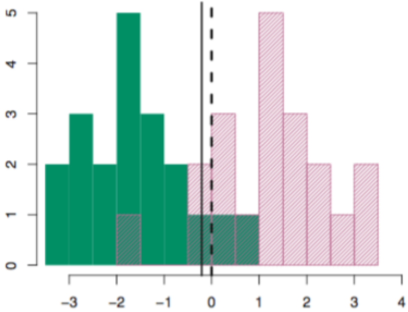
$$P(Y = k|X = x) \quad = \quad \frac{\pi_k \cdot f_k(x)}{\sum_{l=1}^{K} \pi_l \cdot f_l(x)}$$

# The Mechanics II

1. $f_k(x)$ is assumed to be a normal distribution with $\mu_k = \frac{\sum x_{i,k}}{n_k}$ and $\sigma = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_k=k} (x_i - \mu_k)^2$
2. compute for each $k$: $\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + log(\pi_k)$
3. Classify $i$ to be in $k$ if $\delta_k(x) > \delta_j(x) \forall j \neq k$

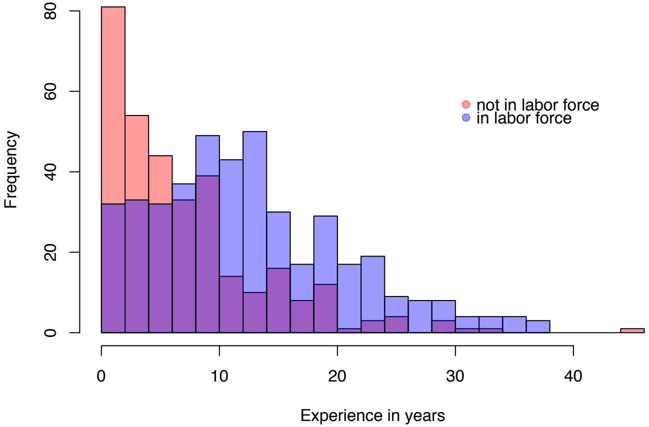
# Simple case: K=2




(James et al, 2013: 140)

# Example: Female Labor Force



# LDA: Female Labor Force Example

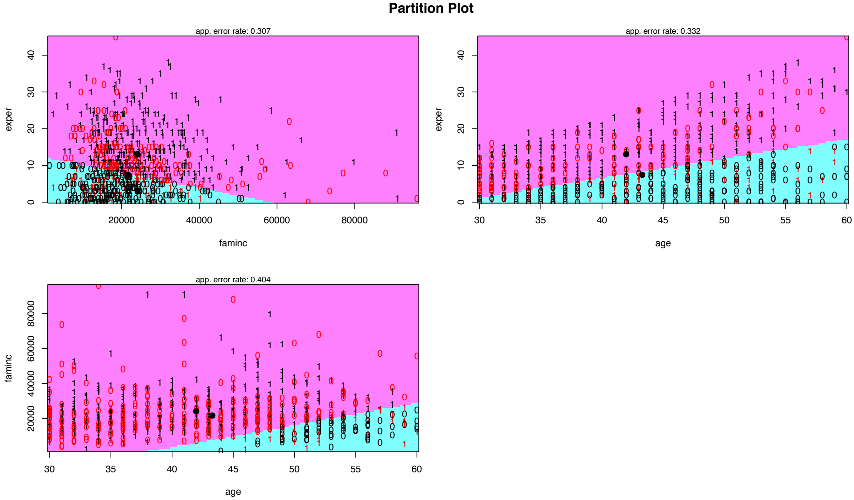
```
> fit <- lda(inlf ~ exper, data=data1, na.action="na.omit", CV=TRUE)
> fit$class
  [1] 1 0 1 0 0 1 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 1 1 0 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 0 1 0 1 0 0 1 1 1 0 1 1
 [78] 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1
[155] 0 1 0 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 0 1 1 1 0 1 0 0 1 1 1
[232] 1 1 0 1 0 0 1 1 1 1 1 1 0 1 0 0 1 1 1 0 0 1 0 1 1 0 1 0 1 0 1 1 1 1 0 1 0 0 0 1 1 1 0 0 1 1 0 1 1 1 0 1
[309] 1 1 1 1 1 1 1 0 1 0 1 1 1 0 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 0 1 1
[386] 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 1 1 1 0 0 0 1 0 1 0 1 1 0
[463] 1 0 0 1 0 1 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 1 0 1 1 1 0 0 0 0 1 1 0 0 1 0
[540] 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 1 0 1 1 1 1 1 0 1 0 1 1 0 0 1 0 1 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
[617] 1 0 1 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 1 1 0
[694] 0 1 0 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 1 1 0 1 1 1 1 0 0 1 1 0 0 1 0 0 1 1
Levels: 0 1
> table(fit$class)

  0   1
315 438
> table(fit$class, data1$inlf)

      0   1
  0 196 119
  1 129 309
```

# Example with several variables

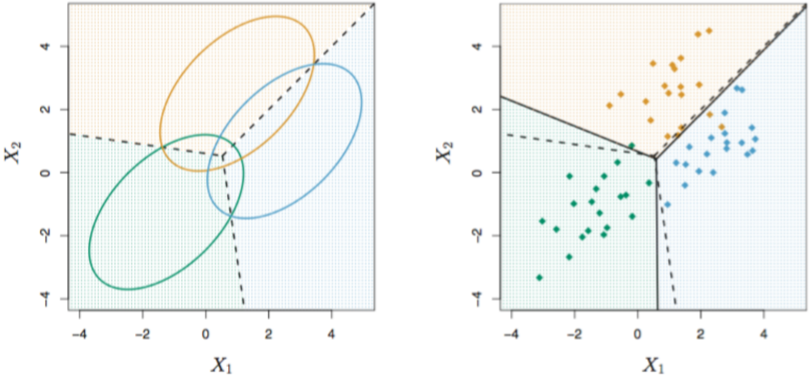
```
> # several variables LDA
> fit <- lda(inlf ~  age + exper + faminc, data=data1, na.action="na.omit", CV=TRUE)
> fit$class
  [1] 1 1 1 0 1 1 1 1 1 1 1 1 0 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 1 0 0 1 0 1 0 0 1 1 1 1 1
 [78] 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 0 1 0
[155] 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 1 1
[232] 1 1 0 1 1 0 1 1 1 1 1 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 1 1 1 1 1
[309] 0 1 1 1 1 1 1 0 1 0 1 1 1 0 0 1 1 0 0 0 1 1 1 1 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 1 0 0 1 1 1 0 1 1 1 1 0 0 1
[386] 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 0 1 1 1 1 1 1 1 0 0 1 0 1 1 1 0 1 0 1 1 1 1 0 1 1 0 1 0 1 0 0
[463] 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1 0 0 0 0
[540] 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 0 0 0 1 0 0 0 1 0 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 0 1 1 0 0 0
[617] 1 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 0 1 0 1 0 1 0 0 0 0 0 1 1 0 1 0 1 0
[694] 0 1 0 1 1 1 0 0 1 1 1 1 1 0 0 1 0 1 1 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 1 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0 0 1
Levels: 0 1
> table(fit$class)

  0   1
309 444
> table(fit$class, data1$inlf)

      0   1
  0 197 112
  1 128 316
```

```
partimat(as.factor(inlf) ~  exper + faminc + age, data=data1, method="lda", nplots.vert=2, nplots.hor=2)
```

# 3 Variables (...and ugliest plot possible)



# K=3 and two variables




(James et al, 2013: 143)

# LDA Summary

- Bayes' rule can help for classification
- But we normally do not know $f_k(x)$ and hence assume normal function and estimate $\mu_k$ and $\sigma$ based on data
- This method is very similar to naive Bayes classifier (which assumes off-diagonal of vcov to be 0)
- Extension of LDA is QDA (Quadratic Discriminant Analysis), more flexible (more data since QDA estimates $\mathbf{\Sigma}_k$ for each $k$)

## Comparison of Classification Methods

# Various Methods

- KNN
- Logistic regression
- LDA
- QDA

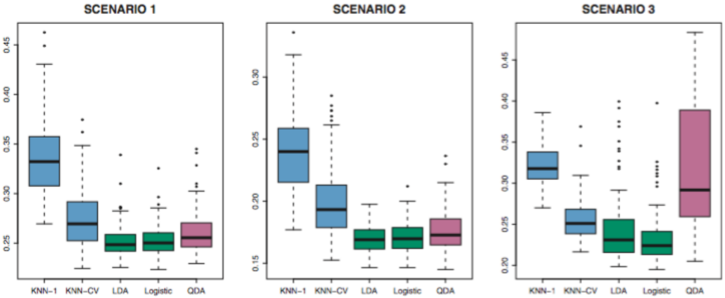
From most structure to least structure:

- Logistic regression/LDA $>>$ QDA $>>$ KNN

Interpretability:

- Logistic regression $>>>$ LDA, QDA, KNN

# Comparison



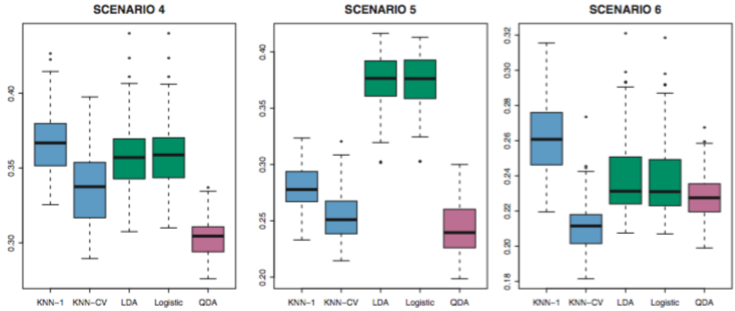

$x_{1i} \sim N(\mu_1, \sigma)$
$x_{2i} \sim N(\mu_2, \sigma)$
$\rho_{x_1, x_2} = 0$

$x_{1i} \sim N(\mu_1, \sigma)$
$x_{2i} \sim N(\mu_2, \sigma)$
$\rho_{x_1, x_2} = -0.5$

$x_{1i} \sim t_1$
$x_{2i} \sim t_2$

(James et al, 2013: 152)

# Comparison




$x_{1i} \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$
$x_{2i} \sim N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$
$\rho_{x_{11}, x_{12}} = 0.5$ but
$\rho_{x_{21}, x_{22}} = -0.5$

$P(k = 2) =$
$\Delta(X_1^2 + X_2^2 + X_1 \cdot X_2)$

$P(k = 2) = f(X_1, X_2)$, whereas $f(x)$ is highly non-linear

(James et al, 2013: 152)

# Summary

- Various classification methods.
- Trade-off between structure and flexibility.
- Every problem has another optimal method.