Warm up - The Desbois Case study

Your NAME

Table of contents

0.1 Quarto 1
0.2 Running Code 1
0.3 Desbois case study 1

0.1 Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

0.2 Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).

0.3 Desbois case study

Loading Desbois data and first glimpse at it:

```r
library(tidyverse)
# package used to import spss file
library(foreign)

don_desbois <- read.spss("presentation_data/Agriculture Farm Lending/desbois.sav", to.data
glimpse(don_desbois)
```

```
Rows: 1,260
Columns: 30
$ CNTY    <fct> Eure, Eure, Eure, Eure, Eure, Eure, Eure, Eure, Eure, Eure, Eu~
$ DIFF    <fct> healthy, healthy, healthy, healthy, healthy, healthy, healthy,~
$ STATUS  <fct> Entreprise individuelle, Entreprise individuelle, Entreprise i~
$ HECTARE <dbl> 166, 101, 138, 166, 137, 107, 100, 69, 176, 177, 108, 123, 87,~
$ ToF     <fct> cereals, cereals, dairy farm, cereals, mixed livestock, cereal~
$ OWNLAND <fct> yes, no, yes, yes, yes, yes, no, yes, yes, yes, yes, yes, yes,~
$ AGE     <dbl> 35, 35, 42, 50, 33, 45, 34, 30, 44, 39, 40, 40, 40, 52, 49, 44~
$ HARVEST <fct> 1988, 1988, 1988, 1988, 1988, 1988, 1988, 1988, 1988, 1988, 19~
$ r1      <dbl> 0.449, 0.450, 0.332, 0.363, 0.440, 0.306, 0.717, 0.566, 0.145,~
$ r2      <dbl> 0.622, 0.617, 0.819, 0.733, 0.650, 0.755, 0.320, 0.465, 0.881,~
$ r3      <dbl> 0.25500, 0.24110, 0.55680, 0.35960, 0.31420, 0.26350, 0.16280,~
$ r4      <dbl> 0.11450, 0.10840, 0.18510, 0.13050, 0.13820, 0.08055, 0.11680,~
$ r5      <dbl> 0.334, 0.341, 0.147, 0.232, 0.302, 0.225, 0.601, 0.499, 0.115,~
$ r6      <dbl> 0.785, 0.518, 0.700, 0.773, 0.846, 0.709, 0.894, 0.863, 0.439,~
$ r7      <dbl> 0.585, 0.393, 0.310, 0.495, 0.580, 0.523, 0.748, 0.761, 0.348,~
$ r8      <dbl> 0.20020, 0.12500, 0.38950, 0.27790, 0.26580, 0.18700, 0.14550,~
$ r11     <dbl> 0.6628, 0.7098, 0.4142, 0.4661, 0.7715, 0.8178, 0.6598, 0.6619~
$ r12     <dbl> 1.3698, 1.2534, 0.6370, 1.0698, 1.4752, 1.4682, 1.1018, 1.2360~
$ r14     <dbl> 0.23200, 0.14970, 0.48470, 0.37350, 0.25630, 0.18610, 0.18070,~
$ r17     <dbl> 0.0884, 0.0671, 0.0445, 0.0621, 0.0489, 0.0243, 0.0352, 0.0879~
$ r18     <dbl> 0.0694, 0.0348, 0.0311, 0.0480, 0.0414, 0.0173, 0.0315, 0.0758~
$ r19     <dbl> 0.1660, 0.1360, 0.1030, 0.1080, 0.1270, 0.0652, 0.1290, 0.2400~
$ r21     <dbl> 0.1340, 0.0802, 0.0890, 0.0851, 0.0838, 0.0390, 0.0784, 0.1630~
$ r22     <dbl> 0.3219, 0.3133, 0.2945, 0.1909, 0.2567, 0.1472, 0.3211, 0.5170~
$ r24     <dbl> 0.295, 0.365, 0.166, 0.265, 0.257, 0.191, 0.322, 0.305, 0.180,~
$ r28     <dbl> 0.475, 0.434, 0.350, 0.475, 0.475, 0.443, 0.401, 0.464, 0.475,~
$ r30     <dbl> 0.35000, 0.29780, 0.24680, 0.37590, 0.36690, 0.37590, 0.27240,~
$ r32     <dbl> 0.4313, 0.3989, 0.3187, 0.4313, 0.4313, 0.4257, 0.3697, 0.3886~
$ r36     <dbl> 0.886, 0.351, 1.300, 1.385, 0.886, 1.316, 0.441, 0.761, 2.147,~
$ r37     <dbl> 0.572, 0.867, 0.475, 0.470, 0.520, 0.431, 0.803, 0.656, 0.359,~
```

Replacing for convenience `DIFF` target by factor `Y` with `0` (healthy) or `1` (failing):

```r
don_desbois <- don_desbois %>%
    mutate(Y = as.factor(if_else(DIFF=='healthy', 0, 1)), DIFF = NULL,
           .before = everything())
```

Your turn to play

Using the data set at hand, try to retrieve some findings of the Desbois case study.

You might need `R` packages `FactoMineR`, `ROCR` and `MASS::lda` function to perform the tasks ("YOUR CODE HERE") described in the following slides.

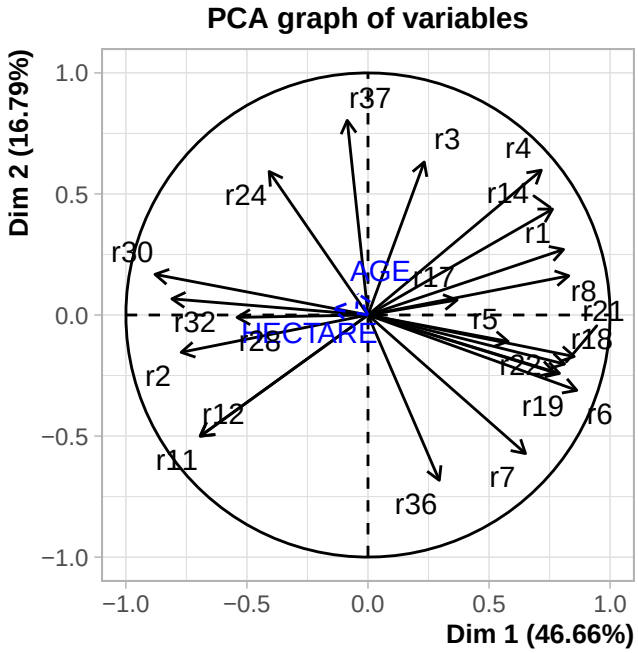
Each time I show an example of what is expected.

I suggest that you start using `Quarto` (setup here) to code and track/publish your results. It will be requested for your projects (it is very similar to `RMarkdown`). It integrates perfectly with `RStudio`.

Principal Component Analysis (PCA) of financial ratios

First perform PCA on financial ratios (use for example package FactoMineR), then display correlations between ratios and the two first principal components :

```r
########## YOUR CODE HERE ######################
```



To be compared with article Figure 1

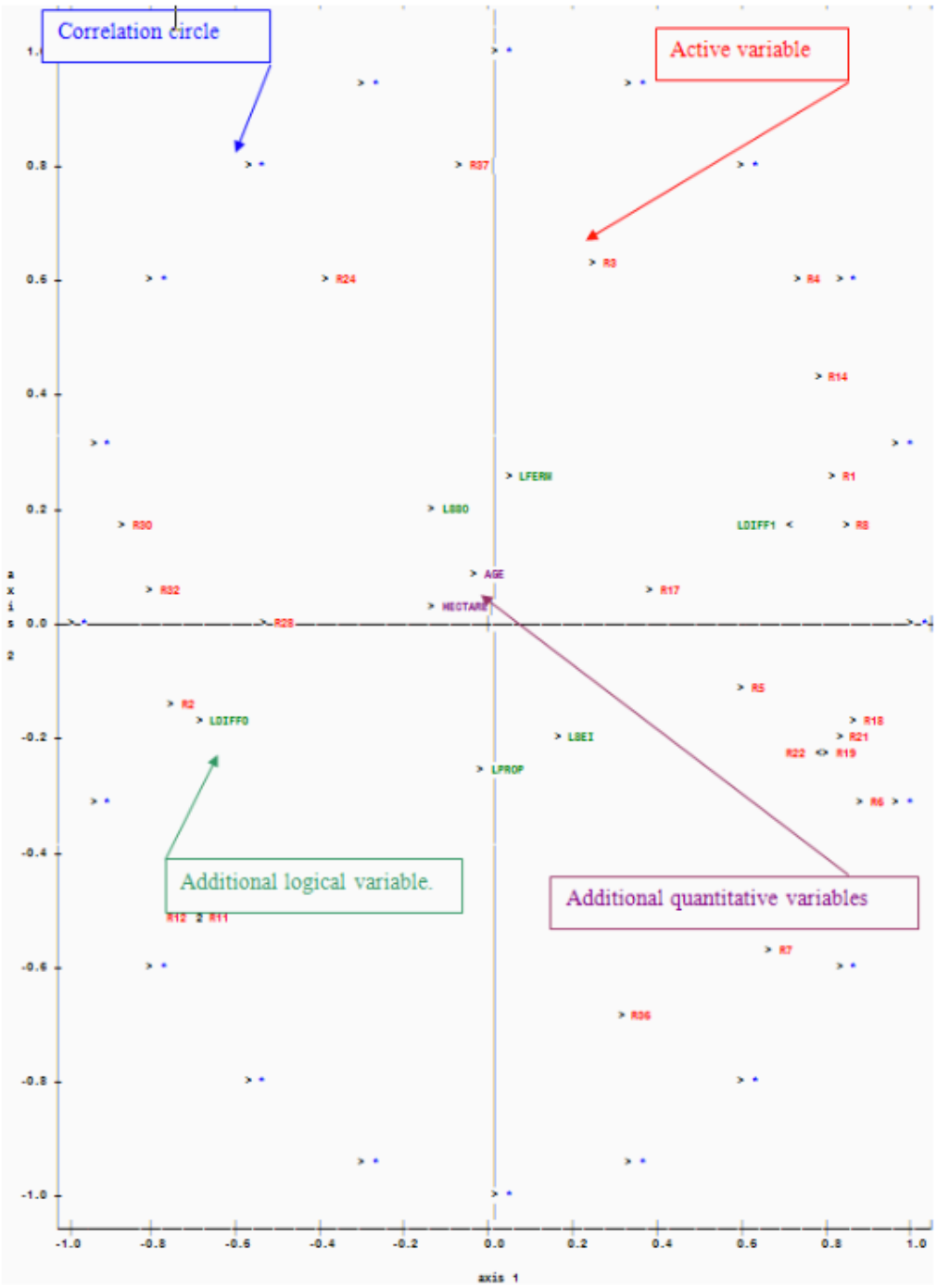
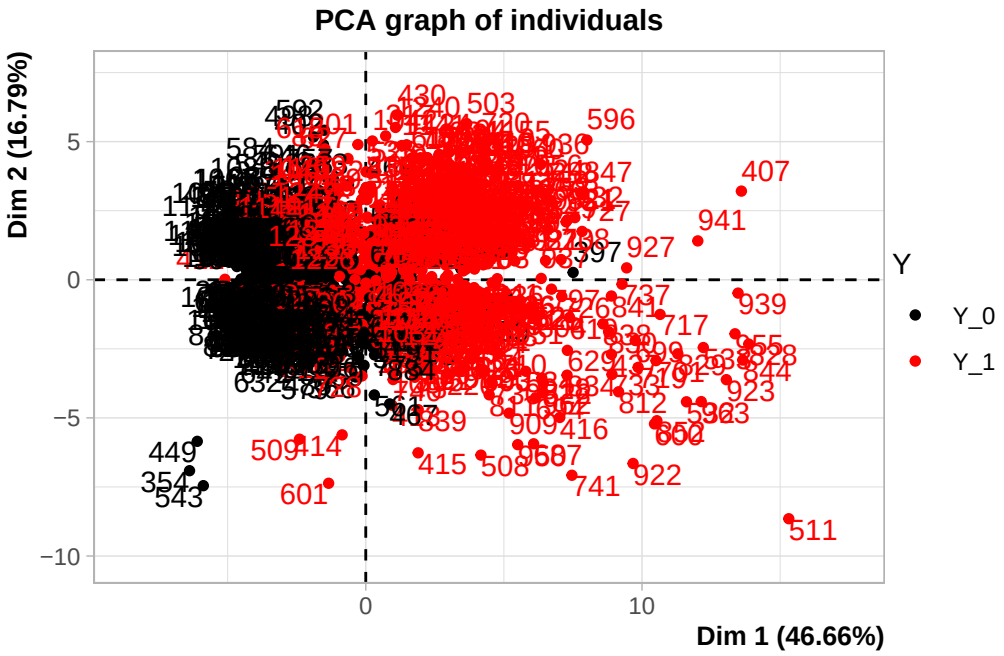


Figure 1. Projection of the financial ratios on the first factorial plane of the normalized PCA.

Following Desbois path, visualize the farm holdings in data set as a bivariate plot on the first two components of PCA (based on financial ratios), use variable Y (0="healthy"; 1="failing") to colour your observations:

```
########## YOUR CODE HERE ######################
```



To be compared with article Figure 2

It strikes Desbois that the PCA (even if not a discriminative/scoring procedure), seems to do a rather good job identifying defaulting farms on the training set.

Disclaimer: in general it won't be necessarily the case as PCA operates only on the predictors without "knowledge" of target variable.

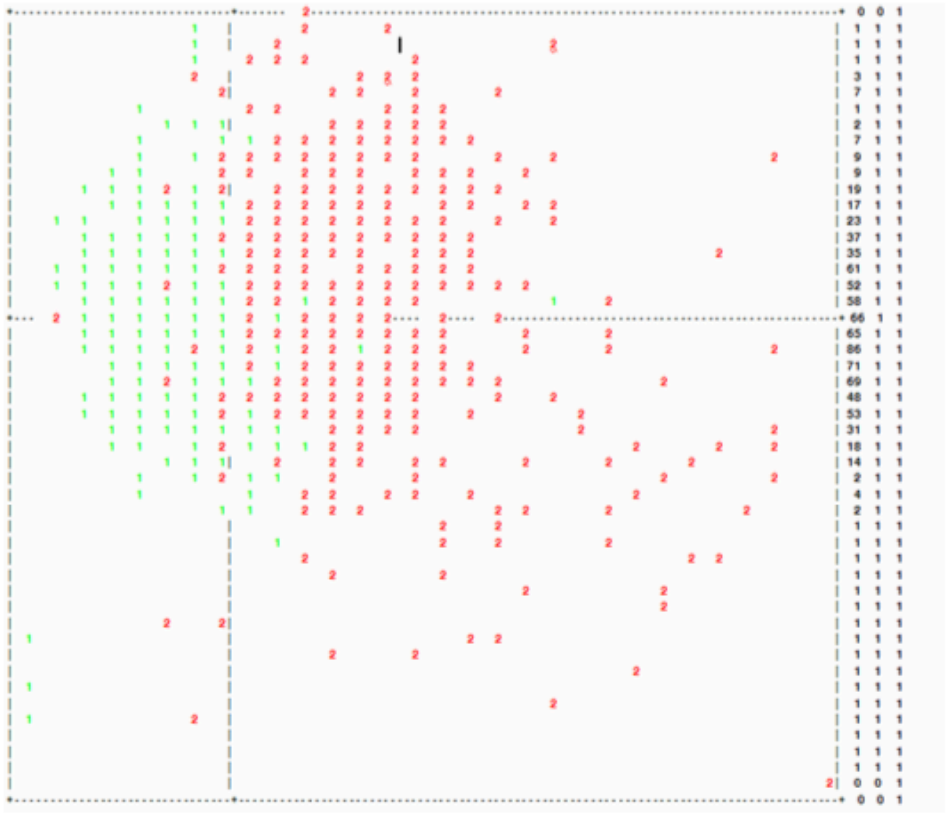


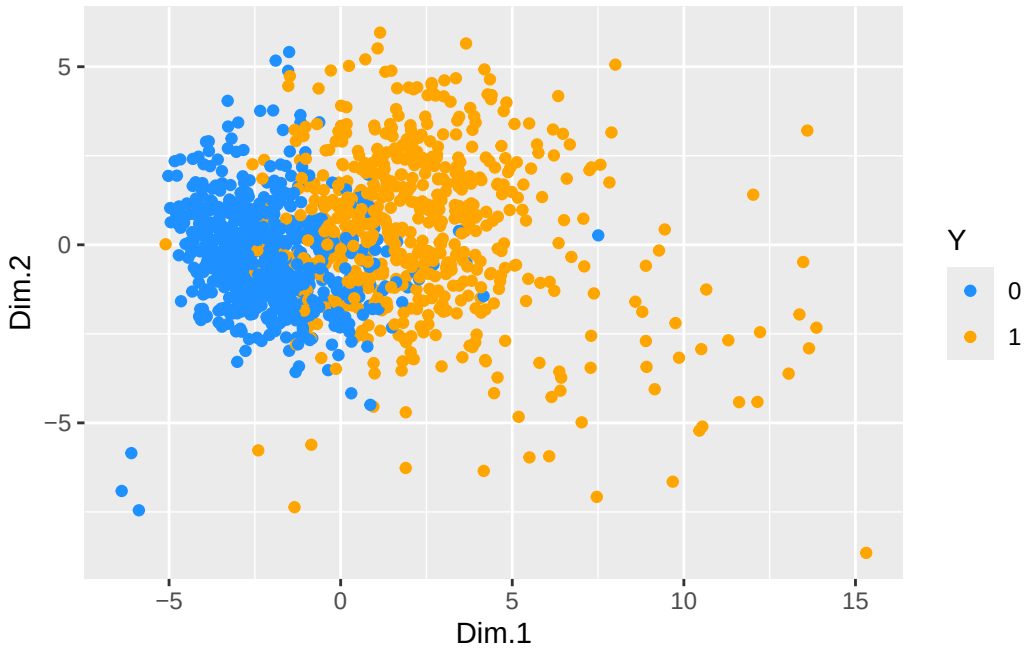
Figure 2. Plot of the farm holdings in the first factorial plane of the normalized PCA based on financial ratios with illustrative variable LDIFF (1="healthy"; 2="failing").

As suggested by Desbois, extract the first principal component (some help here), and use it as a Scoring function.

Today it will allow us to use a first Scoring function without introducing the Logistic Regression that you have not yet studied in class.

```
########## YOUR CODE HERE ######################
```

Below the score for each observation is the x-axis or first principal component:



Looking at this plot Desbois concludes that a simple classifier can be devised using the first PCA coordinate as a classifier (setting a threshold at $PC_1 > 0.02$):

inertia. Considering the respective positions of the healthy and failing groups of farm holdings on the first factorial plane, we see that the *F1* axis could be used as an index of classification (see Figures 2 and 4).

Indeed, the average coordinate of the "healthy" group on the *F1* axis is approximately equal to $\mu_0 \approx -0.6754$ while the average coordinate of the "failing" group is approximately equal to $\mu_1 \approx 0.7266$.

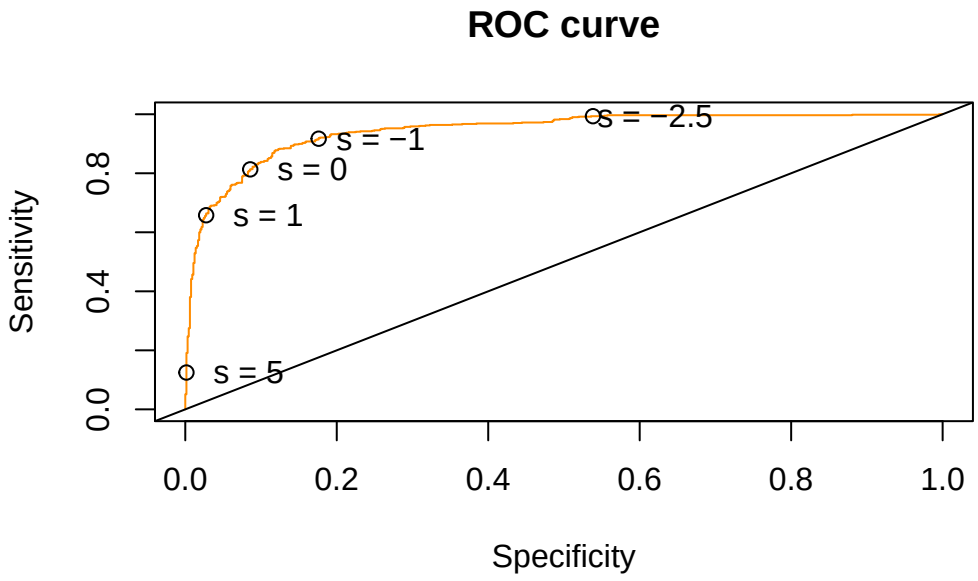
A geometrical rule of classification for a farm holding i_0 which does not belong to the training sample consists in positioning the farm holding in relation to the "pivot point", i.e. the median point of the segment linking the two group barycenters on the *F1* axis, that is to say

$$\frac{\mu_0 + \mu_1}{2} = 0.0256 :$$

Going further than Desbois and for illustrative purposes only, use PC_1 as a very naive Scoring function.

Plot below the ROC curve as defined in the Scoring part of the presentation (you can use package `ROCR`, but a simple for-loop for different cutoff values s will do the job):

```
########## YOUR CODE HERE ######################
```



We follow closely the case study, but we will see that ROC curves are usually evaluated on a hold-out data set.

Then compare with a Scoring function obtained with Linear Discriminant Analysis (LDA). For a reminder on LDA see for example Hastie et al. (2009, Chapter 4.3 LDA, p. 103-111).

You can use Desbois variables selected in the article by a stepwise procedure (using Wilk's lambda[1]). Selected variables are:

[1]We won't describe the procedure here. A SPSS script is given in the article. It is not straightforward to reproduce it with R. I tried scripting manually the procedure and it works but is not very interesting.

Table 6. Stepwise selection of the predictors discriminating the two groups of farm holdings according to the criterion minimizing Wilks' lambda.

Step	Predictors Included	Wilks Lambda		Exact F			
		Statistic	dof 1	Statistic	dof 1	dof 2	P level
1	r1	0.580	1	909.310	1	1258	0.000
2	r32	0.502	2	624.223	2	1257	0.000
3	r14	0.467	3	478.407	3	1256	0.000
4	r17	0.453	4	378.131	4	1255	0.000
5	r2	0.445	5	312.664	5	1254	0.000
6	r3	0.437	6	268.833	6	1253	0.000
7	r36	0.429	7	237.793	7	1252	0.000
8	r21	0.423	8	212.917	8	1251	0.000
9	r7	0.422	9	190.439	9	1250	0.000
10	r18	0.419	10	173.322	10	1249	0.000
11		0.420	9	192.062	9	1250	0.000

Notes: For Wilks' Lambda, dof2 = 1, dof3 = 1258. In step 11, r1 was excluded.

For instance, the total debt rate [*r1*] ratio, introduced in the first step because acting as the most discriminating variable, is eliminated in the last step of the selection process because it can be sufficiently reconstituted in an approximate way as a linear combination of the other ratios introduced into the preceding steps.

From this follows the linear discriminant function *D1* (Fisher's score), which is expressed with "standardized coefficients" as a linear combination of the standardized discriminating variables (see Table 7).

So basically you can fit LDA with `r2+r3+r7+r14+r17+r18+r21+r32+r36` as predictors (I use `R` formula notation to ease your copy-paste).

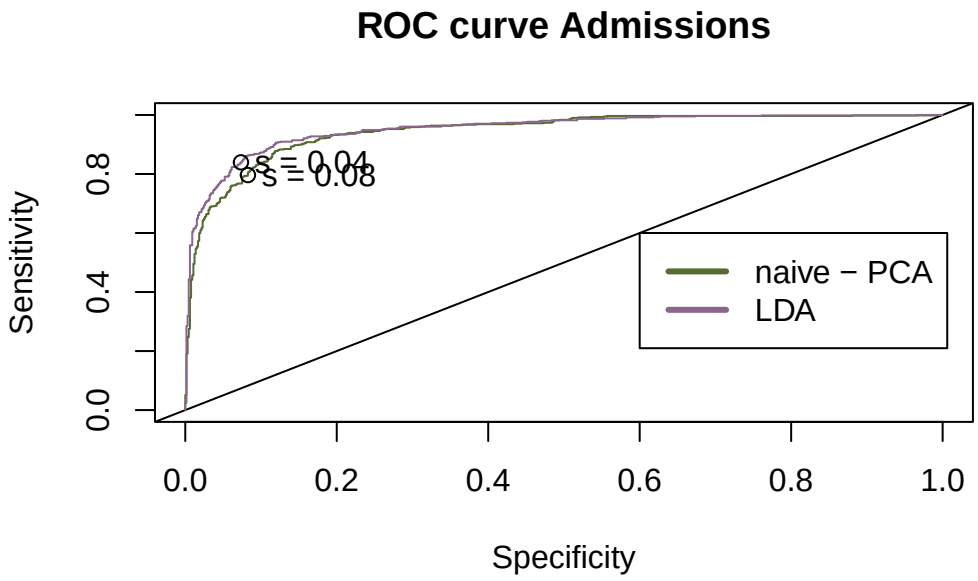
Fit LDA (`MASS::lda`) with model specification of your choice. Then display model coefficients:

```
########## YOUR CODE HERE ######################
```

```
       LD1
r2  -2.542
r3   2.398
r7   1.099
r14  0.604
r17 18.702
r18 -6.798
r21 -0.622
r32 -4.163
r36  0.192
```

Now compare the LDA Scoring function with the naive Score build with first component of PCA using a ROC curve (on training set):

```
########## YOUR CODE HERE ######################
```



As a very soft and intuitive introduction to logistic regression. First discretize a ratio r of your choice.

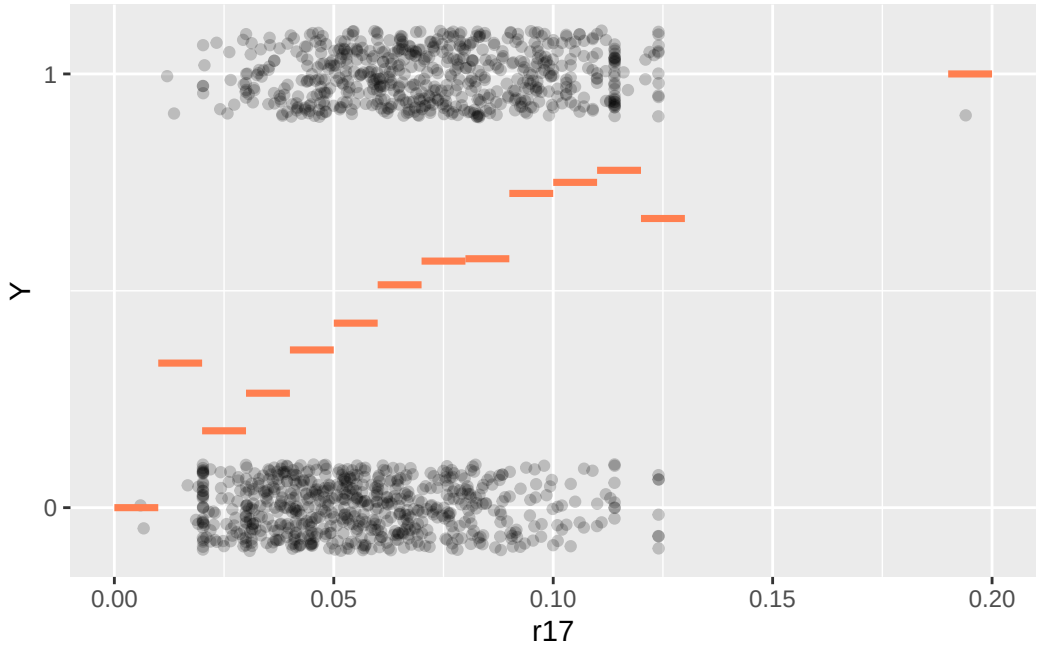
- Compute proportion of defaults among each class. For example using 0.01 steps for ratio $r17$:

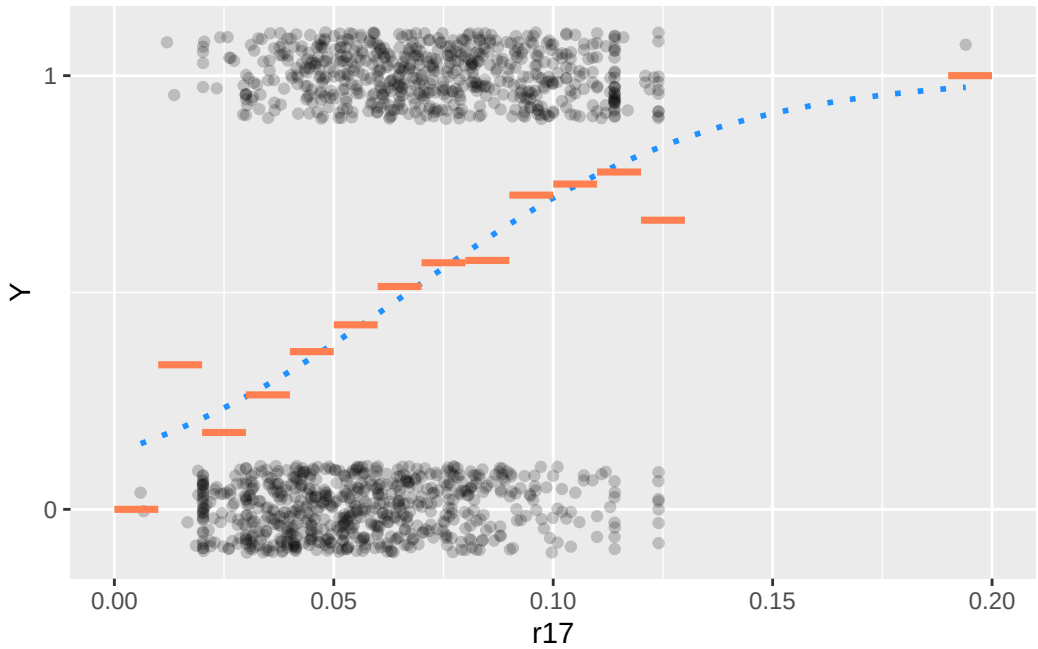
```
########## YOUR CODE HERE ######################
```

r17_class	0	1	Mean(Y)
[0,0.01)	2	0	0.0000
[0.01,0.02)	4	2	0.3333
[0.02,0.03)	65	14	0.1772
[0.03,0.04)	106	38	0.2639
[0.04,0.05)	112	64	0.3636
[0.05,0.06)	104	77	0.4254
[0.06,0.07)	88	93	0.5138
[0.07,0.08)	63	83	0.5685
[0.08,0.09)	49	66	0.5739
[0.09,0.1)	27	71	0.7245
[0.1,0.11)	14	42	0.7500
[0.11,0.12)	12	42	0.7778
[0.12,0.13)	7	14	0.6667
[0.19,0.2]	0	1	1.0000

- Then using this table, plot an empirical conditional distribution of default given r classes:

```
########## YOUR CODE HERE ######################
```





This roughly corresponds to the intuitive introduction to the logistic regression model given in Hosmer et al. (2013) using a Coronary Heart Disease (CHD) event as Y and AGE as X:

The column labeled "Mean" in Table 1.2 provides an estimate of $E(Y|x)$. We assume, for purposes of exposition, that the estimated values plotted in Figure 1.2 are close enough to the true values of $E(Y|x)$ to provide a reasonable assessment of the functional relationship between CHD and AGE. With a dichotomous outcome variable, the conditional mean must be greater than or equal to zero and less than or equal to one (i.e., $0 \leq E(Y|x) \leq 1$). This can be seen in Figure 1.2. In addition, the plot shows that this mean approaches zero and one "gradually". The change in the $E(Y|x)$ per unit change in x becomes progressively smaller as the conditional mean gets closer to zero or one. The curve is said to be *S-shaped* and resembles a plot of the cumulative distribution of a continuous random variable. Thus, it should not seem surprising that some well-known cumulative distributions have been used to provide a model for $E(Y|x)$ in the case when Y is dichotomous. The model we use is based on the logistic distribution.

Table 1.2 Frequency Table of Age Group by CHD

Age Group	n	Coronary Heart Disease		Mean
		Absent	Present	
20–29	10	9	1	0.100
30–34	15	13	2	0.133
35–39	12	9	3	0.250
40–44	15	10	5	0.333
45–49	13	7	6	0.462
50–54	8	3	5	0.625
55–59	17	4	13	0.765
60–69	10	2	8	0.800
Total	100	57	43	0.430

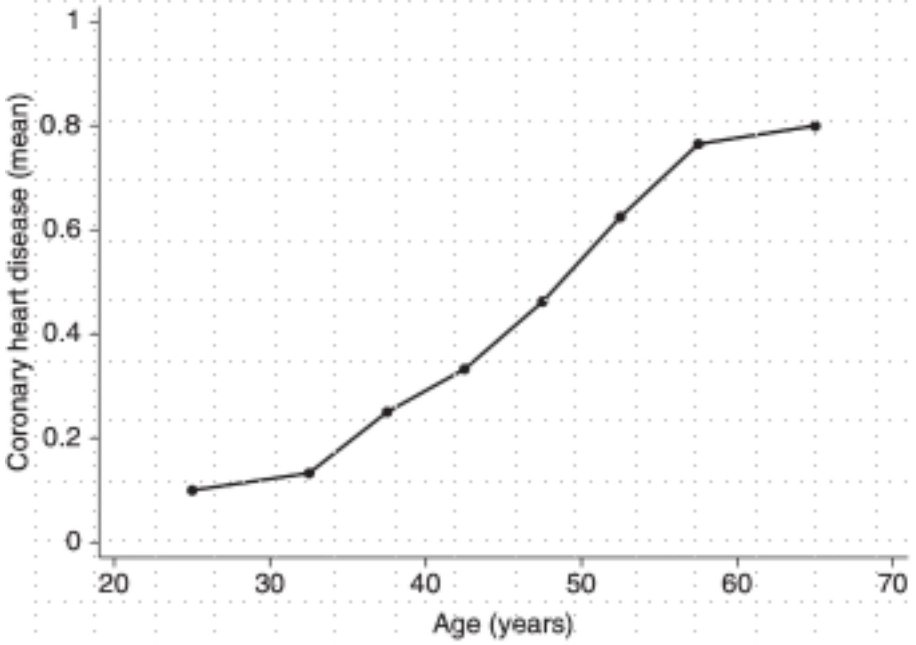


Figure 1.2 Plot of the percentage of subjects with CHD in each AGE group.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. Springer New York.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression. In *Wiley Series in Probability and Statistics*.