Scoring

M2 D3S/EGR 2025-2026

Louis Olive

louis.olive@gmail.com / louis.olive@ut-capitole.fr

September 16, 2025

Outline

Outline

- (Warm Up) Desbois case study - solutions
- Scoring - reminders
- The Logistic Regression model

Desbois case study

Desbois case study

Financial problems of farm holdings

The article from Desbois (2008) (available here together with a sample data set) shows a complete case study around the detection of financial risks applicable to farm holdings.

Among others, Linear Discriminant Analysis and Logistic Regression (plus stepwise variable selection procedure) are used.

ROC curves are then provided to compare the two methods.

The original data set uses a format specific to the SPSS software, but is readable from R using the `foreign` package.

The case study by Desbois will be partly reproduced today and in the next lessons to quickly apply scoring techniques.

Desbois case study

Loading Desbois data and first glimpse at it:

Code

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

don_desbois <- read.spss("presentation_data/Agriculture Farm Lending/desbois.sav", to.data.frame = TRUE) %>% as_tibble()
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,…
```

Desbois case study

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

▾ Code

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

We give in the following slides the definitions of Desbois ratio $r_1, \ldots, r_{37}$, that we will use today, as found in the article.

They might give you inspiration to create new predictors/features for the project to come.

Desbois case study

Capitalization ratios

- r1 total debt / total assets;
- r2 stockholders' equity / invested capital;
- r3 short term debt / total debt;
- r4 short term debt / total assets;
- r5 long and medium term debt / total assets;

Desbois case study

Weight of the debt ratios

- r6 total debt / gross product;
- r7 long and medium term debt / gross product;
- r8 short term debt / gross product;

Desbois case study

Liquidity ratios

- r11 working capital / gross product;
- r12 working capital / (real inputs - financial expenses);
- r14 short term debt / circulating assets;

Desbois case study

Debt servicing ratios

- r17 financial expenses / total debt;
- r18 financial expenses / gross product;
- r19 (financial expenses + refunding of long and medium term capital) / gross product;
- r21 financial expenses / EBITDA;
- r22 (financial expenses + refunding of long and medium term capital) / EBITDA;

Desbois case study

Capital profitability ratio

- r24 EBITDA / total assets;

Desbois case study

Earnings ratios

- r28 EBITDA / gross product;
- r30 available income / gross product;
- r32 (EBITDA - financial expenses) / gross product;

Desbois case study

Productive activity ratios

- r36 immobilized assets / gross product;

- r37 gross product / total assets.

Desbois case study

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`.

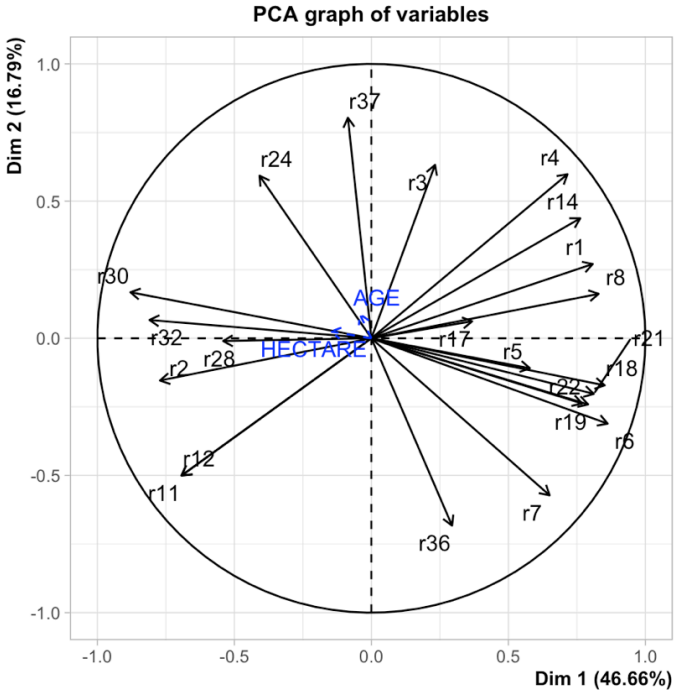
Desbois case study

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 :

▾ Code

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

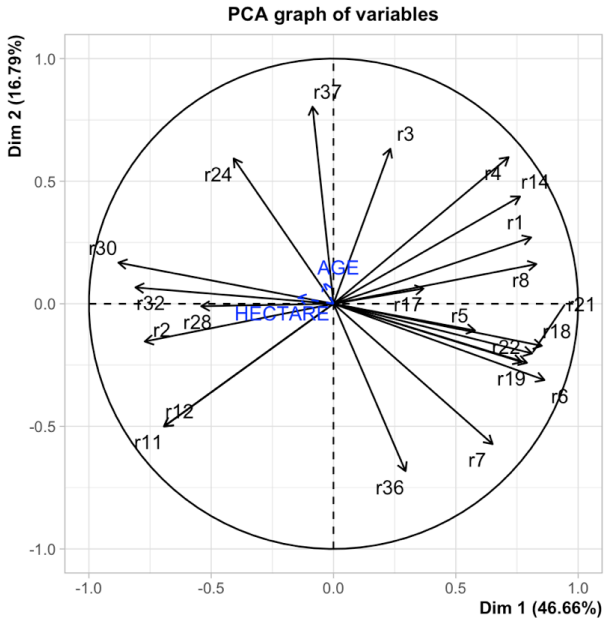


Desbois case study

Principal Component Analysis (PCA) of financial ratios

▾ Code

```
# More details here http://factominer.free.fr/factomethods/principal-components-analysis.html
res.pca = FactoMineR::PCA(don_desbois,
                          scale.unit = TRUE,
                          quanti.sup = c(4, 7), # HECTARE / AGE excluded from Desbois analysis
                          quali.sup = c(1, 2, 3, 5, 6, 8),
                          ncp = 5,
                          graph=FALSE)
plot(res.pca, choix = "var")
```



Desbois case study

To be compared with article Figure 1

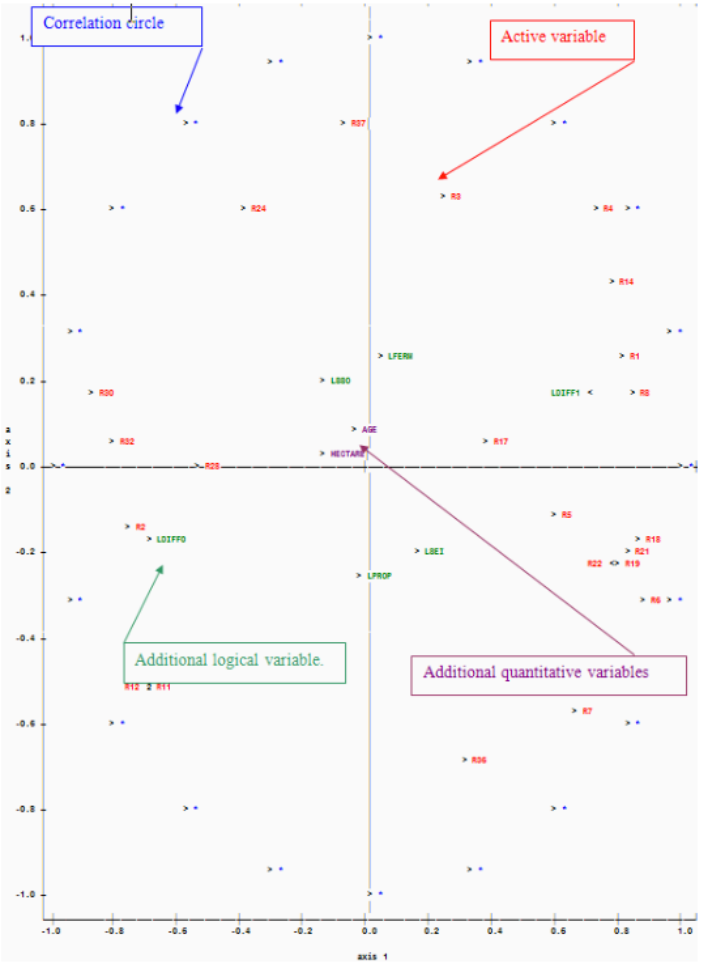


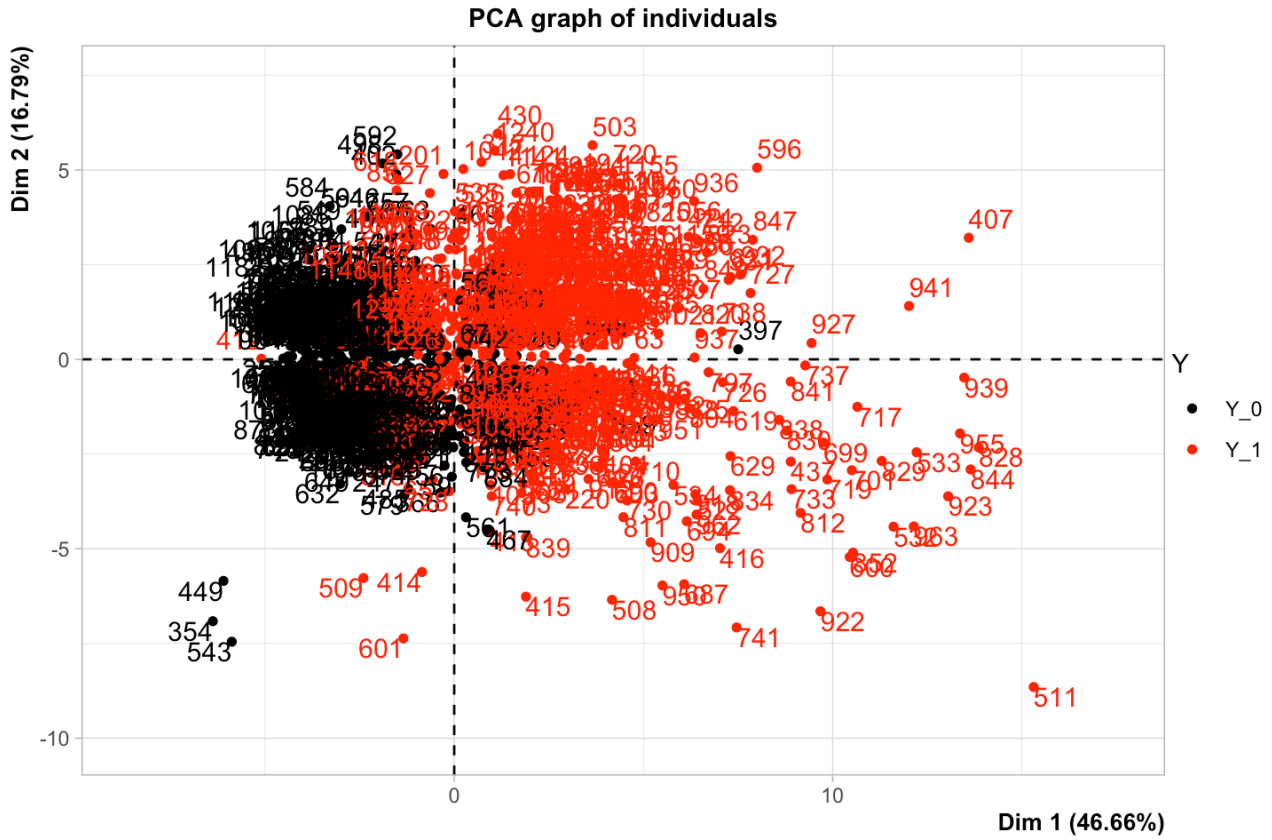
Figure 1. Projection of the financial ratios on the first factorial plane of the normalized PCA.

Desbois case study

Code

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

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:



Desbois case study

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.

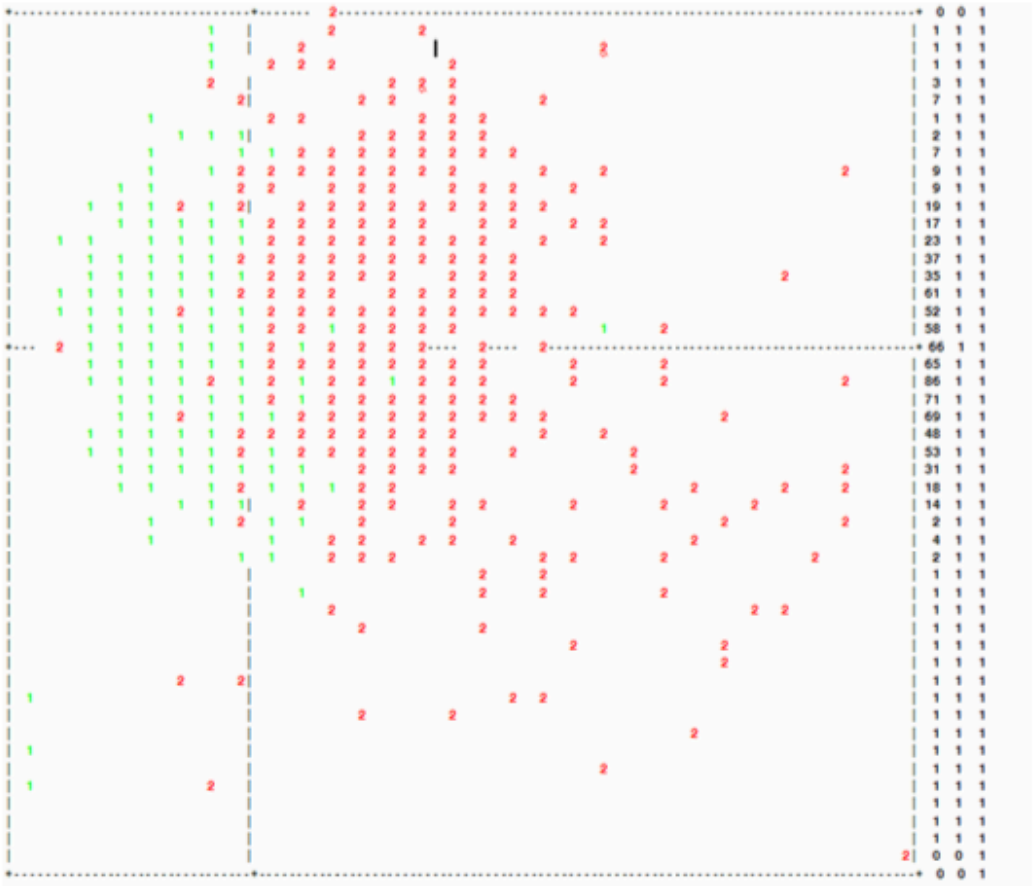


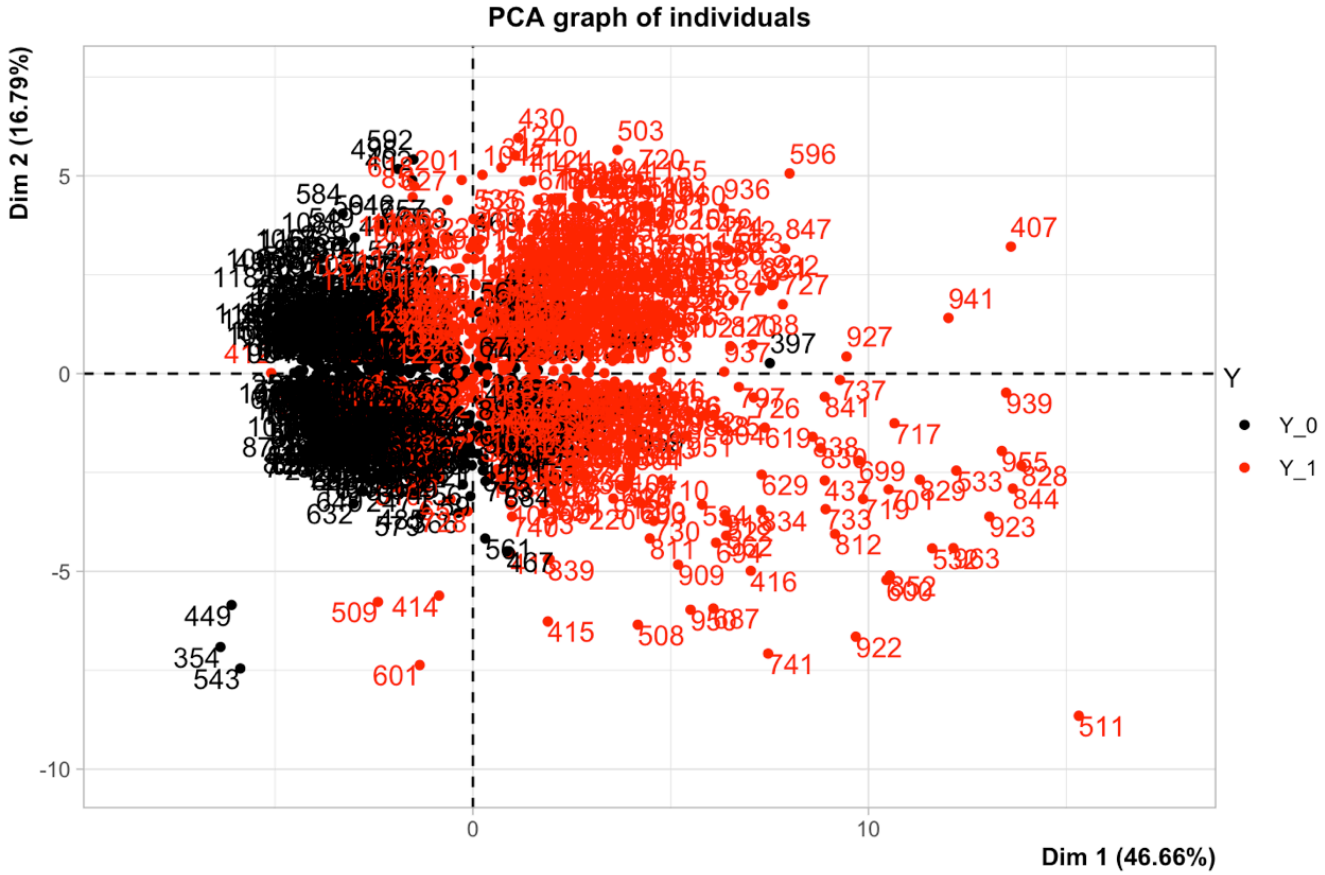
Figure 2. Plot of the farm holdings in the first factorial plane of the normalized PCA based on financial ratios with illustrative variable LDIFF (1="healthy"; 2="failing").

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

Desbois case study

Code

```
# Similar to Desbois Fig 2.
# Plot of the farm holdings in the first factorial plane of the normalized PCA based on financial ratios
# with illustrative variable Y (0="healthy"; 1="failing")
FactoMineR::plot.PCA(res.pca, axes=c(1, 2), choix="ind", habillage=1, invisible = c("quali"))
```



Desbois case study

Looking at the preceding plot Desbois concludes that a simple classifier can be devised using the first PCA coordinate (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

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

Desbois case study

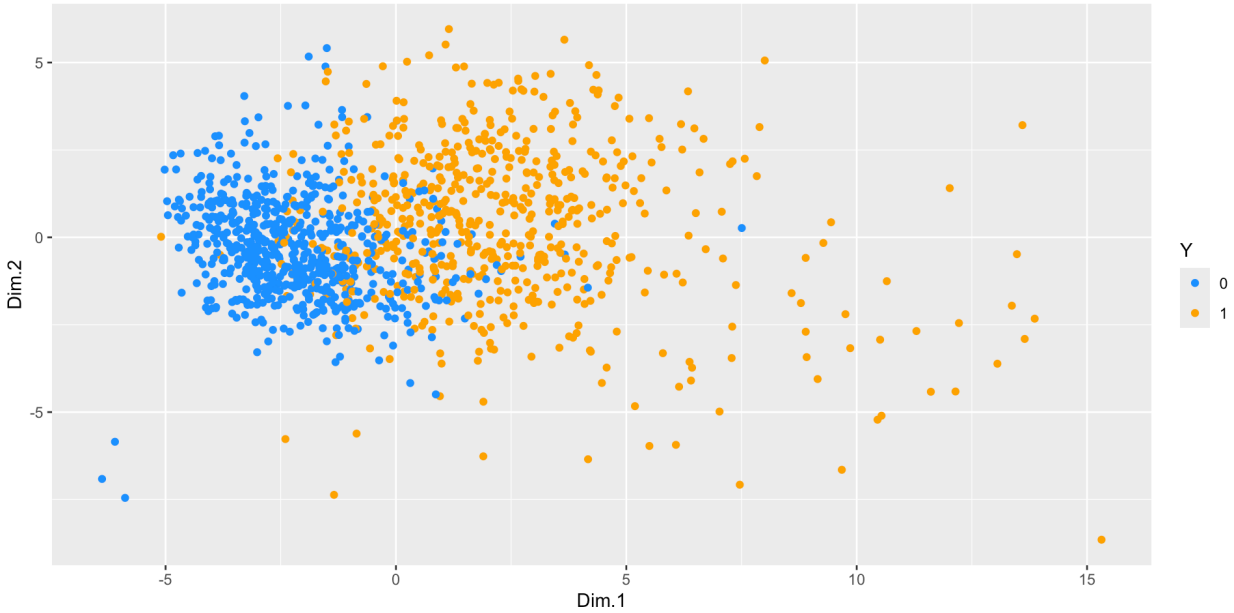
As suggested by Desbois, extract the first principal component (some help here), and use it as a Scoring function.

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

Code

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

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



Desbois case study

We show below how to obtain the first five Principal components (Desbois only uses the first two):

Code

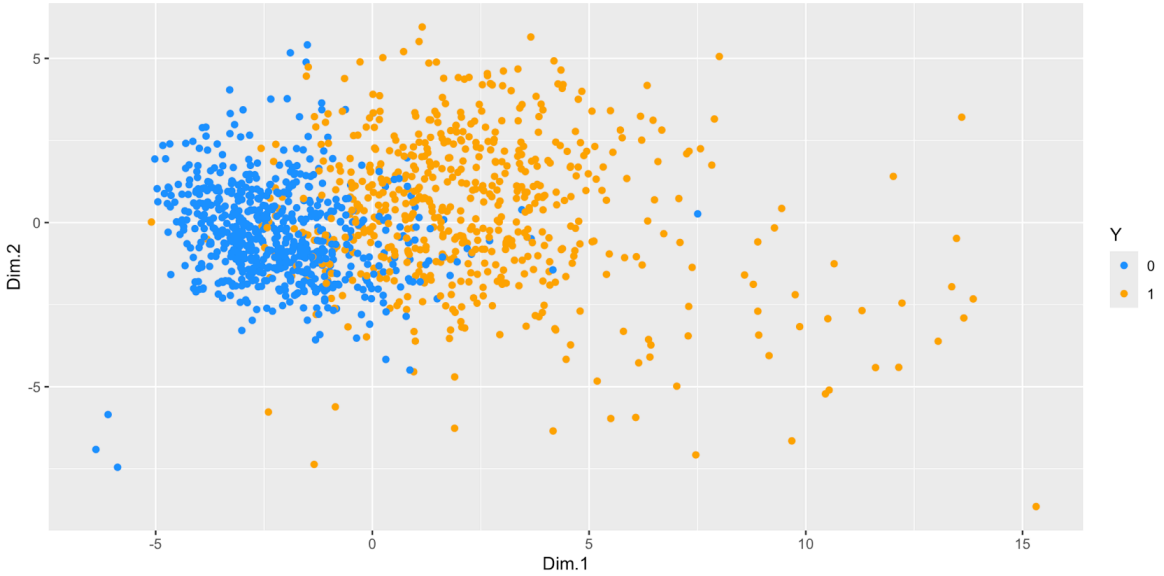
```
# https://stats.stackexchange.com/questions/460787/pcr-after-pca-with-mixed-data-how-to-extract-export-the-pcs-as-new-variables-i
# https://stats.stackexchange.com/questions/494866/how-to-explain-the-numerical-discrepancy-between-factominerpca-and-the-svd
# https://stats.stackexchange.com/questions/134282/relationship-between-svd-and-pca-how-to-use-svd-to-perform-pca

# Extracting first five Principal components from FactomineR object
# Using FactomineR 'notation' (U,sv,V)
# (X nxp matrix of Desbois observations (only r1...r37), SVD decomposition X = U %*% diag(vs) %*% t(V),
# where U: unitary matrix of left-singular vectors, vs: singular values, V: unitary matrix of left-singular vectors)
# SVD object from FactomineR res.pca$svd

# Principal components X %*% V = U %*% diag(vs) %*% t(V) %*% V = U %*% diag(vs)
principal_components_1 <- res.pca$svd$U %*% diag(res.pca$svd$vs[1:5])

# Can also be directly extracted from:
principal_components_2 <- res.pca$ind$coord

# Or using PC = X %*% V (where X has been centered and scaled
# base R 'scale' uses a different scaling factor than FactomineR hence sqrt(n / n-1) correction)
X_for_PCA <- don_desbois %>% select(r1:r37) %>% as.matrix()
principal_components_3 <- sqrt(nrow(X_for_PCA) / (nrow(X_for_PCA) - 1)) * scale(X_for_PCA) %*% res.pca$svd$V
# perform svd using base R on scaled matrix
s <- svd(scale(X_for_PCA))
# correct for negative signs
s$v[, c(2, 3, 4, 5)] <- s$v[, c(2, 3, 4, 5)] * -1
V <- s$v[,1:5]
principal_components_4 <- sqrt(nrow(X_for_PCA) / (nrow(X_for_PCA) - 1)) * scale(X_for_PCA) %*% V
```

Desbois case study

Below we define the score for each observation $\boldsymbol{x}$ as the first principal component $\boldsymbol{PC_1(x)}$ (x-coordinate below, with $\boldsymbol{PC_2(x)}$ as y-coordinate):

▾ Code

```
naive_score <-  bind_cols(don_desbois, as_tibble(principal_components_2))
ggplot(naive_score) +
    geom_point(aes(x = Dim.1, y = Dim.2, col = Y)) +
    scale_colour_manual(values = c("dodgerblue", "orange"))
```



Defaulting farms or observations $\boldsymbol{x}$ appear in orange, healthy in blue.

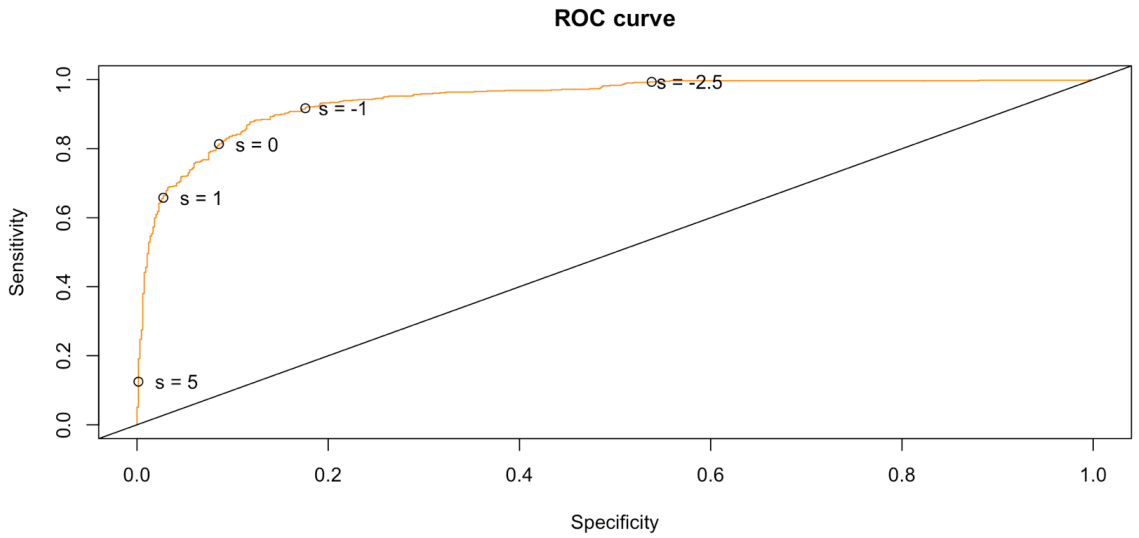
Desbois case study

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):

▾ Code

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

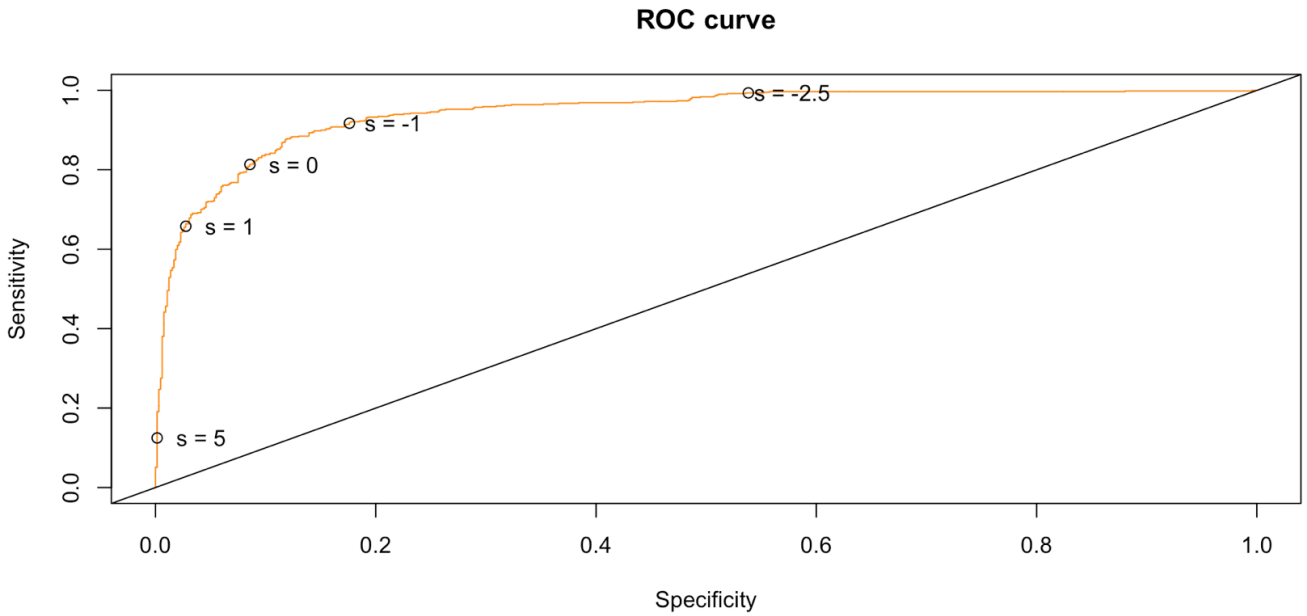


We follow closely the case study, but we will see that ROC curves are usually evaluated on a hold-out data set.

Desbois case study

Code

```
library("ROCR")

# naive score
pred <- prediction(naive_score$Dim.1, naive_score$Y)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf, main="ROC curve", xlab="Specificity",
     ylab="Sensitivity", col = "darkorange",
     print.cutoffs.at = c(-2.5,-1,0.00,1,5),
     cutoff.label.function = function(x) {paste0("              s = ",round(x,2))})
abline(0, 1) #add a 45 degree line
```



Desbois case study

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:

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 [r*1*] 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).

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.

Desbois case study

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

▾ Code

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

Using `r2+r3+r7+r14+r17+r18+r21+r32+r36`, we obtain:

```
       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
```

Similar[1] to what Desbois obtains:

Table 9. Unstandardized coefficients issued from the original variables, to be used for the score computation.

Predictors	Function D1
stockholders' equity / invested capital [r2]	-2.542
short-term debt / total debt [r3]	2.398
long and medium-term debt / gross product [r7]	1.099
short-term debt / circulating asset [r14]	0.604
financial expenses / total debt [r17]	18.702
financial expenses / gross product [r18]	-6.798
financial expenses / EBITDA [r21]	-0.622
(EBITDA - financial expenses) / gross product [r32]	-4.163
immobilized assets / gross product [r36]	0.192
(Constant)	-0.444

1. R does not provide constant for LDA which won't affect Scoring function ordering

Desbois case study

Code

```
lda_desbois <- MASS::lda(Y~r2+r3+r7+r14+r17+r18+r21+r32+r36, data=don_desbois)
round(lda_desbois$scaling,3)
```

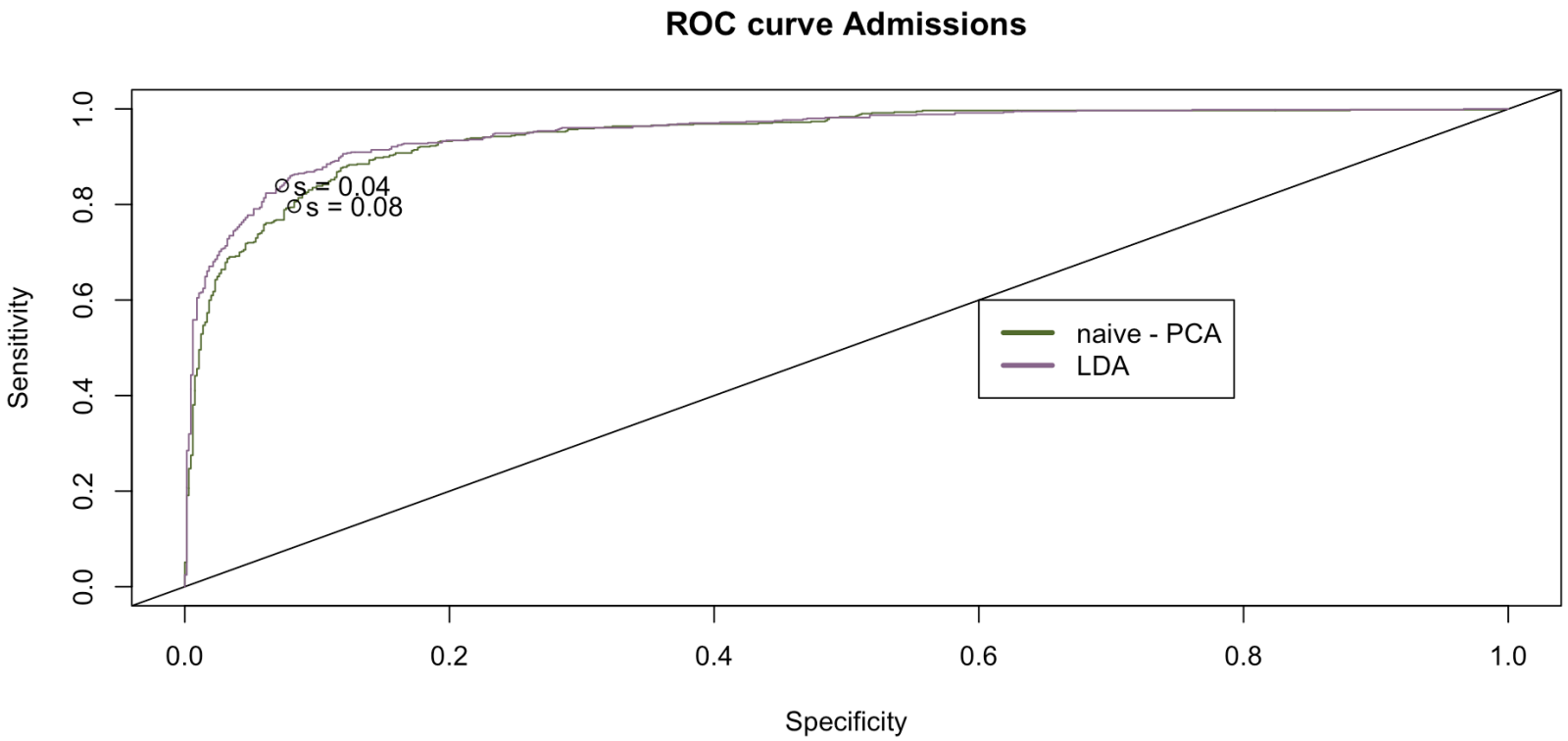
```
       LD1
r2  -2.542
r3   2.398
r7   1.099
r14  0.604
r17 18.702
r18 -6.798
r21 -0.622
r32 -4.163
r36  0.192
```

Desbois case study

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

Code

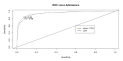
```
1  ########## YOUR CODE HERE ######################
```



Desbois case study

Code

```
# for ROCR
labels_desbois <- don_desbois$Y

# naive score
predictions_naive_score <- naive_score$Dim.1

naive_score_group <- naive_score %>%
  group_by(Y) %>%
  summarize(mean_score = mean(Dim.1), n = n())

cutoff_naive <- mean(naive_score_group$mean_score)

pred <- prediction(predictions_naive_score, labels_desbois)
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf, main="ROC curve Admissions", xlab="Specificity",
     ylab="Sensitivity", col = "darkolivegreen",
     print.cutoffs.at = c(cutoff_naive),
     cutoff.label.function = function(x) {paste0("                s = ",round(x,2))})
abline(0, 1) #add a 45 degree line

# lda
predictions_lda <- predict(lda_desbois)$x

lda_score_group <- bind_cols(don_desbois, predictions_lda) %>%
  group_by(Y) %>%
  summarize(mean_score = mean(LD1), n = n())

cutoff_lda <- mean(lda_score_group$mean_score)
```



Desbois case study

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 $r17$:

▾ Code

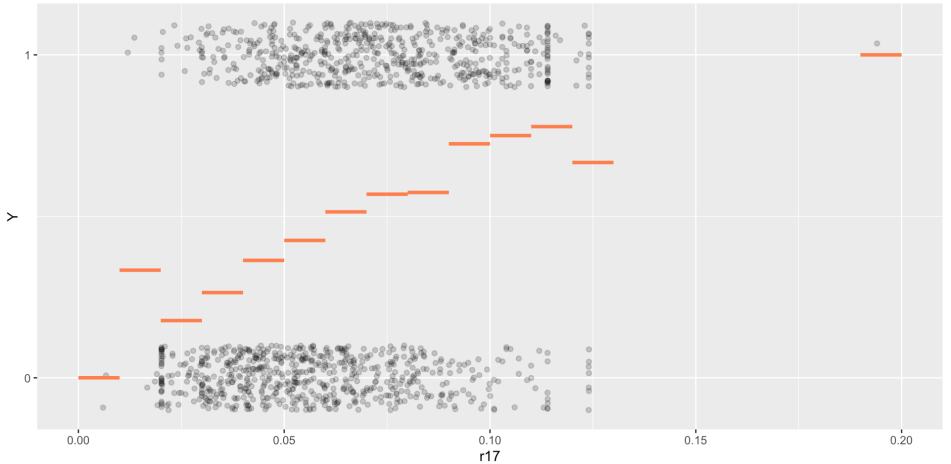
```
########## YOUR CODE HERE ######################
```

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:

▾ Code

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



Desbois case study

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

▾ Code

```
class_width <- 0.01
don_desbois_binned <- don_desbois %>%
    mutate(r17_bins = cut(r17, breaks = seq(0, 0.2, class_width),
                                right = FALSE, dig.lab = 4, include.lowest = TRUE),
           min = floor(r17 / class_width) * class_width,
           max = if_else(r17 == 0 , 1,
                         # customers with 0$ balance should belong to [0, width) class
                         # or be excluded
                         ceiling(r17 / class_width))  * class_width,
           max = if_else(min==max, (ceiling(r17 / class_width) + 1)  * class_width, max)) %>%
    group_by(r17_bins, min, max, Y) %>%
    summarize(n=n()) %>%
    ungroup() %>%
    pivot_wider(names_from = Y, values_from = n) %>%
    replace_na(list(`0` = 0, `1` = 0)) %>%
    mutate(`Mean(Y)` = round(`1` / (`1` + `0`), 4))
cat(simplermarkdown::md_table(don_desbois_binned %>% select(r17_class=r17_bins, `0`, `1`,`Mean(Y)` )))
```

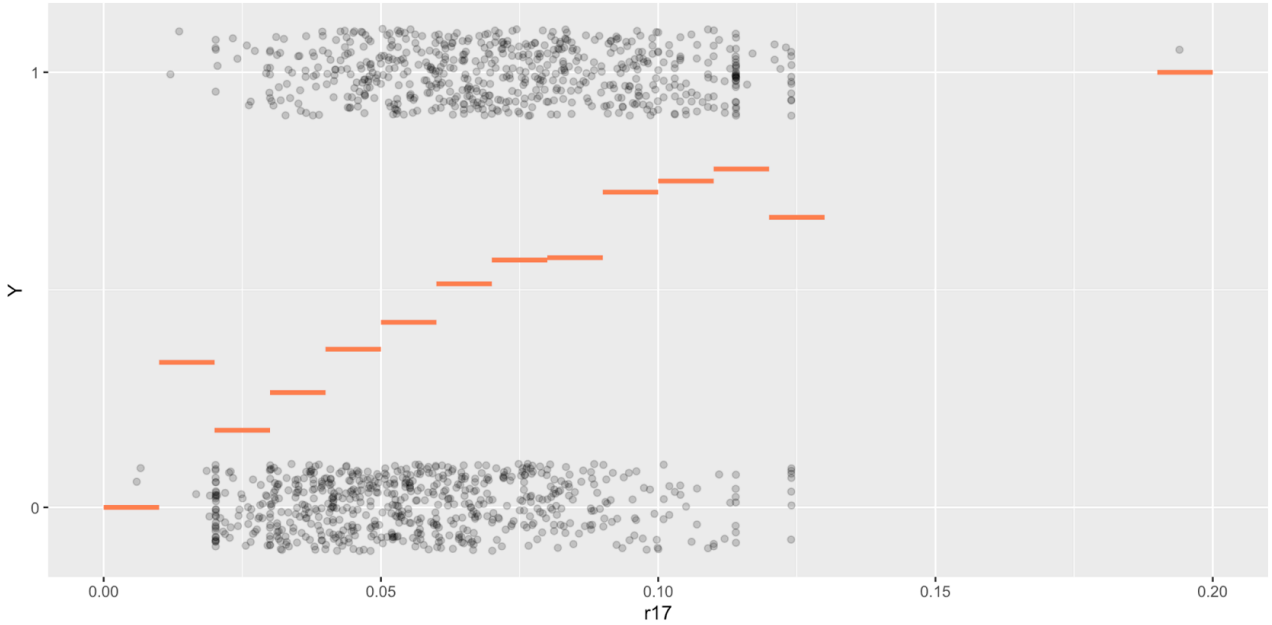
r17_class	0	1	Mean(Y)
[0,0.01)	2	0	0.0000
[0.01,0.02)	4	2	0.3333
[0.02,0.03)	65	14	0.1772
[0.03,0.04)	106	38	0.2639
[0.04,0.05)	112	64	0.3636
[0.05,0.06)	104	77	0.4254
[0.06,0.07)	88	93	0.5138
[0.07,0.08)	63	83	0.5685
[0.08,0.09)	49	66	0.5739
[0.09,0.1)	27	71	0.7245
[0.1,0.11)	14	42	0.7500
[0.11,0.12)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

Desbois case study

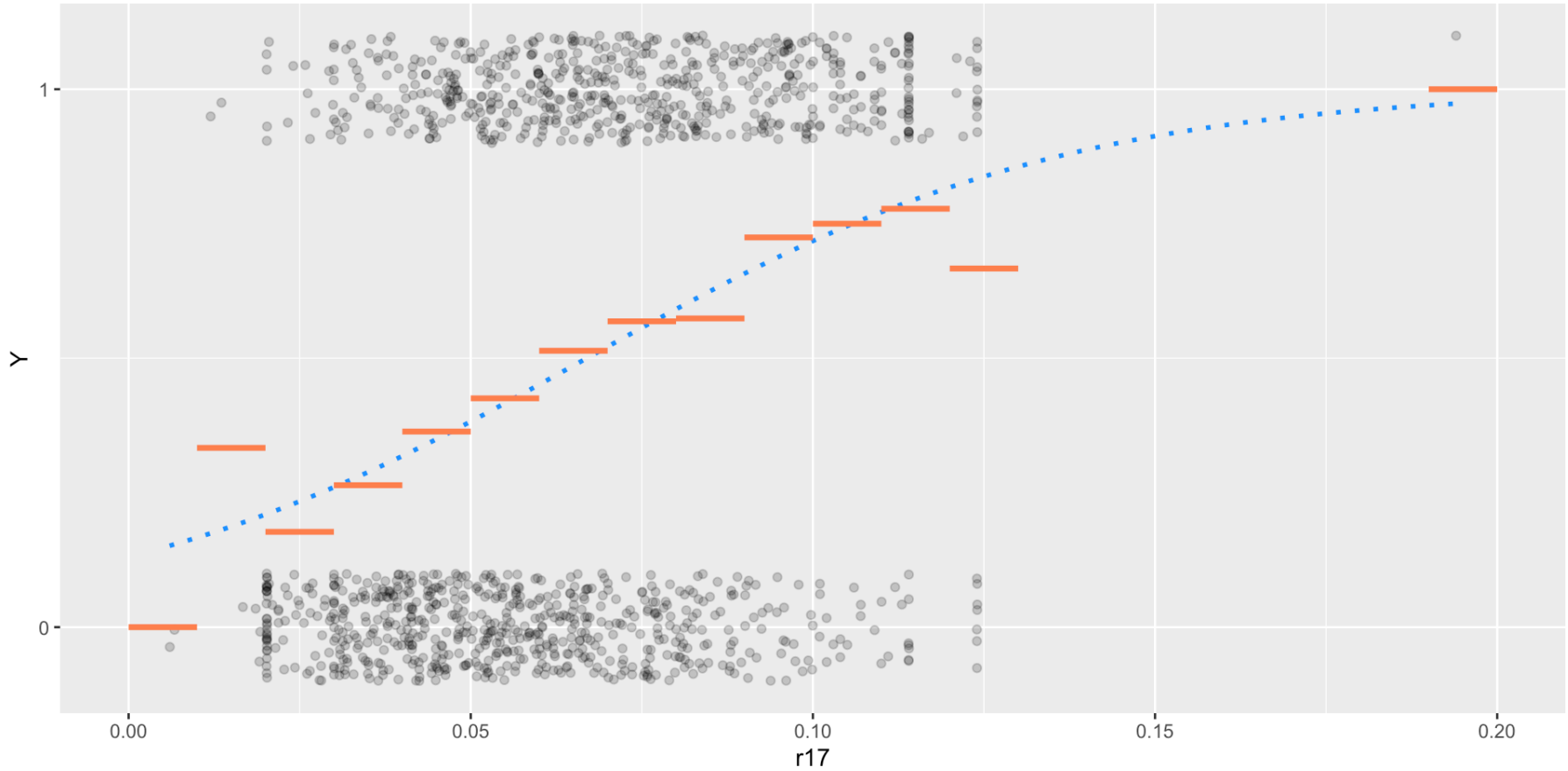
Then using this table, plot an empirical conditional distribution of default given r classes:

▾ Code

```
(default_occurrence <-
 ggplot(don_desbois %>% mutate(Y = if_else(Y == "1", 1, 0)), aes(x=r17, y=Y)) +
 geom_jitter(alpha=0.2, height=0.1) +
 geom_segment(data = don_desbois_binned,
              aes(x = min, xend = max, y = `Mean(Y)`, yend = `Mean(Y)`),
              color = 'coral',
              linewidth=1.25)) +
 scale_y_continuous(breaks = c(0, 1))
```



Desbois case study



We notice that the mean default occurrence with respect to r_{17} classes follows a kind of "S"-shaped curve or **sigmoid** function. Note that the shape depends on classes width and might change. We "fitted" (blue-dotted) $\sigma : x \to \sigma(x) = \frac{e^x}{1+e^x}$ also known as logistic function to the mean default occurences (orange segments).

Desbois case study

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 $\boldsymbol{Y}$ and AGE as $\boldsymbol{X}$:

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: Absent	Coronary Heart Disease: Present	Mean
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

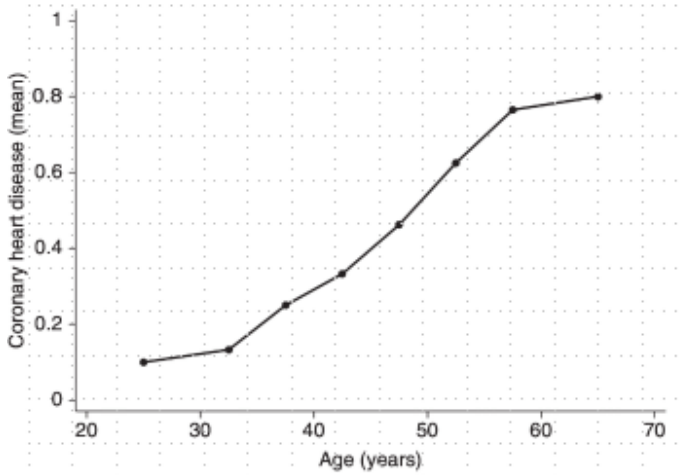
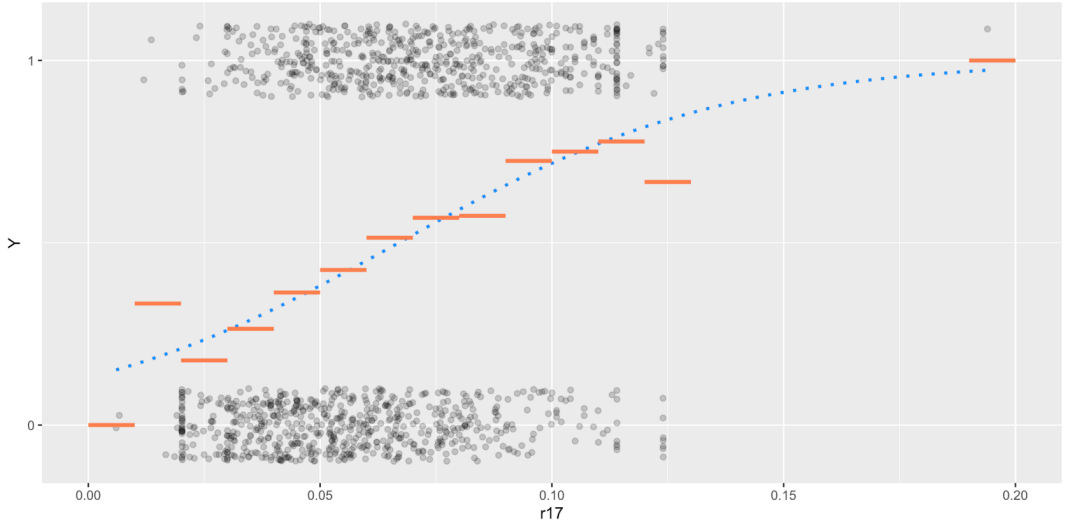


Figure 1.2 Plot of the percentage of subjects with CHD in each AGE group.

Desbois case study

Code

```
(default_occurrence <-
 ggplot(don_desbois %>% mutate(Y = if_else(Y == "1", 1, 0)), aes(x=r17, y=Y)) +
 geom_jitter(alpha=0.2, height=0.1) +
 geom_smooth(method = "glm",
             formula = y ~ x,
             method.args = list(family = "binomial"),
             se = FALSE,
             col = "dodgerblue",
             linetype = "dotted") +
 geom_segment(data = don_desbois_binned,
              aes(x = min, xend = max, y = `Mean(Y)`, yend = `Mean(Y)`),
              color = 'coral',
              linewidth=1.25)) +
 scale_y_continuous(breaks = c(0, 1))
```



Scoring - reminders

Scoring

Supervised learning - Loss

To achieve that goal we try to find a “best” or optimal **classifier** (or prediction function in general):

$$f : \mathrm{X} \to \mathrm{Y}$$

We first must define some criterion to assess the classifier performance (what is optimal?).

For that purpose we consider a **loss** function $\ell : \mathrm{Y} \times \mathrm{Y} \to \mathbb{R}^+$:

$$\begin{cases} \ell(y, z) > 0 & \text{if } y \neq z \\ \ell(y, z) = 0 & \text{if } y = z \end{cases}$$

$\ell(y, f(x))$ is the loss or cost of predicting $f(x)$ while the true output is y.

In the context of binary classification a natural selection for loss is the 0-1 loss:

$$\ell(y, z) = \mathbf{1}_{y \neq z}$$

We will see later in the course that other losses are used.

Scoring

Supervised learning - Risk

We then define the expected **risk** of a prediction function f given the loss ℓ as:

$$\mathrm{R}(f) = \mathbb{E}[\ell(Y, f(X))]$$

In the context of binary classification $Y \in \{0, 1\}$ and 0-1 loss, we have:

$$\mathrm{R}(f) = \mathbb{P}[Y \neq f(X))]$$

Given a data set $(x_i, y_i) \in \mathbf{X} \times \mathbf{Y}$, we also define the empirical risk of a classifier f as:

$$\hat{\mathrm{R}}(f) = \frac{1}{n} \sum_i \ell(y_i, f(x_i))$$

Scoring

Supervised learning - Bayes classifier

We now define the **Bayes(ian) classifier**[1] $f^* : \mathrm{X} \to \mathrm{Y}$ as a function that achieves the minimal expected risk among all possible functions:

$$\operatorname*{argmin}_f \mathrm{R}(f)$$

The Bayes classifier depends on ℓ and $\mathbf{P}_{\mathrm{X} \times \mathrm{Y}}$ which is generally unknown (otherwise the job would be easy).

The Bayes classifier for the 0-1 loss is:

$$f^*(x) = \begin{cases} 1 & \text{if } \eta(x) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

where:

$$\eta(x) = \mathbb{P}[Y = 1 | X = x)] = \mathbb{E}[Y = 1 | X = x)]$$

$\eta(x)$ is called the regression function.

1. It is sometimes designed as oracle in the literature.

Scoring

Supervised learning - Theory vs practice

In practice $\mathbf{P}_{\mathbf{X}\times\mathbf{Y}}$ is unknown. What we have is the learning set. The Bayes classifier minimizes the expected risk but is not a function of the learning set. What to do? Conceptually two approaches are used[1]:

- the "probabilistic" approach: we estimate the regression function $\eta(x)$ and use/plug this estimate "inside" the Bayes classifier (Plugin); within this approach mainly two sub-approaches:
  - the "discriminative" approach: we directly model or make an assumption on $\eta(X) = \mathbf{P}_{\mathbf{Y}|\mathbf{X}}$ and estimate it; for example we can make the assumption that $\eta(x)$ is a function of a particular form.
  - the "generative" approach: we model the conditional distribution $\mathbf{P}_{\mathbf{X}|\mathbf{Y}}$ and use Bayes formula to get: $\eta(X) = \frac{\mathbb{P}[X|Y=1)]\mathbb{P}[Y=1)]}{\mathbb{P}[X]} = \frac{\mathbb{P}[X|Y=1)]\mathbb{P}[Y=1)]}{\mathbb{P}[X|Y=1)]\mathbb{P}[Y=1)]+\mathbb{P}[X|Y=0)]\mathbb{P}[Y=0)]}$
- the optimization or "machine learning" approach: finding f minimizing the empirical risk; optimization algorithm or heuristics are used to estimate f; the 0-1 loss is not convex so usually it is replaced by a convex surrogate loss or upper bound; usually re-sampling methods are used to compute empirical risk (f is trained on a training set and empirical risk evaluated on the testing set, f is chosen to minimize empirical risk).

1. See Sébastien Gadat - TSE - Maths of Deep and Machine Learning course here. Other very good references are Philippe Rigollet - MIT - 18.657: Mathematics of Machine Learning here or Erwan Le Pennec - Polytechnique - Introduction to Machine Learning (M2 MSV) slides here. We will see in another course that in the context of the Logistic Regression the two approaches are strongly linked: it can be seen as a probabilistic|discriminative approach or an optimization approach.

Scoring

Scoring function

As introduced in Cornillon et al. (2019, Chapter 11.6 “Prévision - scoring”):

- The aim of Scoring: find a Scoring function or Score $S : \mathbb{R}^p \to \mathbb{R}$ that is “high” in case $\mathbb{P}[Y = 1|X = x)$ is “high” and “low” in case $\mathbb{P}[Y = 0|X = x)$ is “high”.
- We say that $S(x)$ is the score of the observation $x \in \mathbb{R}^p$.
- The notions of Score and classifier are closely linked. Given a Score S and a cutoff $s \in \mathbb{R}$ we obtain the following classifier:

$$f_s(x) = \begin{cases} 1 & \text{if } S(x) \geq s \\ 0 & \text{otherwise} \end{cases}$$

- The values taken by $S(x)$ are less important than the way they will order a set of observation $x_1, \dots x_n$. For any $\phi : \mathbb{R} \to \mathbb{R}$ bijective and increasing function, we say that the scores S and $\phi \circ S$ are equivalent.

Scoring

First example

We show below a simplified example of a credit scorecard that were typically used by banks to assess the creditworthiness of consumer loans applicants (as shown in (Thomas et al., 2002)):

Table 2.1. *Simple scorecard.*

Residential Status		**Age**	
Owner	36	18–25	22
Tenant	10	26–35	25
Living with parents	14	36–43	34
Other specified	20	44–52	39
No response	16	53+	49

Loan Purpose		**Value of CCJs**	
New car	41	None	32
Second-hand car	33	£1–£299	17
Home improvement	36	£300–£599	9
Holiday	19	£600–£1199	–2
Other	25	£1200+	–17

Scoring

First example

Table 2.1. *Simple scorecard.*

Residential Status		Age	
Owner	36	18–25	22
Tenant	10	26–35	25
Living with parents	14	36–43	34
Other specified	20	44–52	39
No response	16	53+	49

Loan Purpose		Value of CCJs	
New car	41	None	32
Second-hand car	33	£1–£299	17
Home improvement	36	£300–£599	9
Holiday	19	£600–£1199	–2
Other	25	£1200+	–17

- a 35-year-old owner wishing to borrow money for home improvement and that has never had a county court judgement (CCJ) will score $\mathbf{129 = (36 + 25 + 36 + 32)}$,
- a 20-year-old living with his parents borrowing for holiday and having more than £1200 of CCJ will score $\mathbf{38 = (22 + 14 + 19 - 17)}$.

Scoring

The regression function, a Scoring function

We defined above, in the context of supervised learning, the regression function $\eta(x) = \mathbb{P}[Y = 1|X = x)]$
.

It is a natural candidate for a Scoring function.

The related Bayes classifier uses $s = \frac{1}{2}$ as cutoff.

As we have just seen in the last sections, $\eta(x)$ is unknown and we will have to approximate or estimate this regression function using the training set.

Scoring

Evaluating a Score

We first define the **confusion matrix**:

	$f_s(X) = 0$	$f_s(X) = 1$	
$Y = 0$	TN	FP	N
$Y = 1$	FN	TP	P

The confusion matrix is a contingency table which cross-tabulates Y with the predicted one $f_s(X)$.

It can be evaluated on the learning set as well as on the testing set.

The following vocabulary originates from medical applications:

- true positive (TP)
- true negative (TN),
- false positive (FP), also Type I error
- false negative (FN), also Type II error

Scoring

Evaluating a Score

More formally we define for a given cutoff s:

$$\alpha(s) = \mathbf{P}(f_s(X) = 1|Y = 0) = P(S(X) \geq s|Y = 0)$$

and

$$\beta(s) = \mathbf{P}(f_s(X) = 0|Y = 1) = P(S(X) < s|Y = 1)$$

$\alpha(s)$ is called **false positive** rate and $\beta(s)$ **false negative** rate. Similarly **specificity** and **sensitivity** are defined:

$$sp(s) = P(S(X) < s|Y = 0) = 1 - \alpha(s)$$

and

$$se(s) = P(S(X) \geq s|Y = 1) = 1 - \beta(s)$$

Scoring

The Receiver Operating Characteristic (ROC) curve

The ROC (Receiver Operating Characteristic) curve allows to visualize α and β on a same graph for all s allowing to choose the cutoff and compare different Scores.

Precisely, the ROC curve of a Score S is a parametric curve of the variable s:

$$\begin{aligned} ROC: \quad & \mathbb{R} \to [0,1]^2 \\ & s \to (x(s) = \alpha(s), y(s) = 1 - \beta(s)) \end{aligned}$$

For a given Score S, the ROC curve passes through the points $(0,0)$ and $(1,1)$, which corresponds to classifying all observations as 0 ($s \to \infty$) or 1 ($s \to -\infty$).

Scoring

The Receiver Operating Characteristic (ROC) curve

Figure 1 shows the two curves.

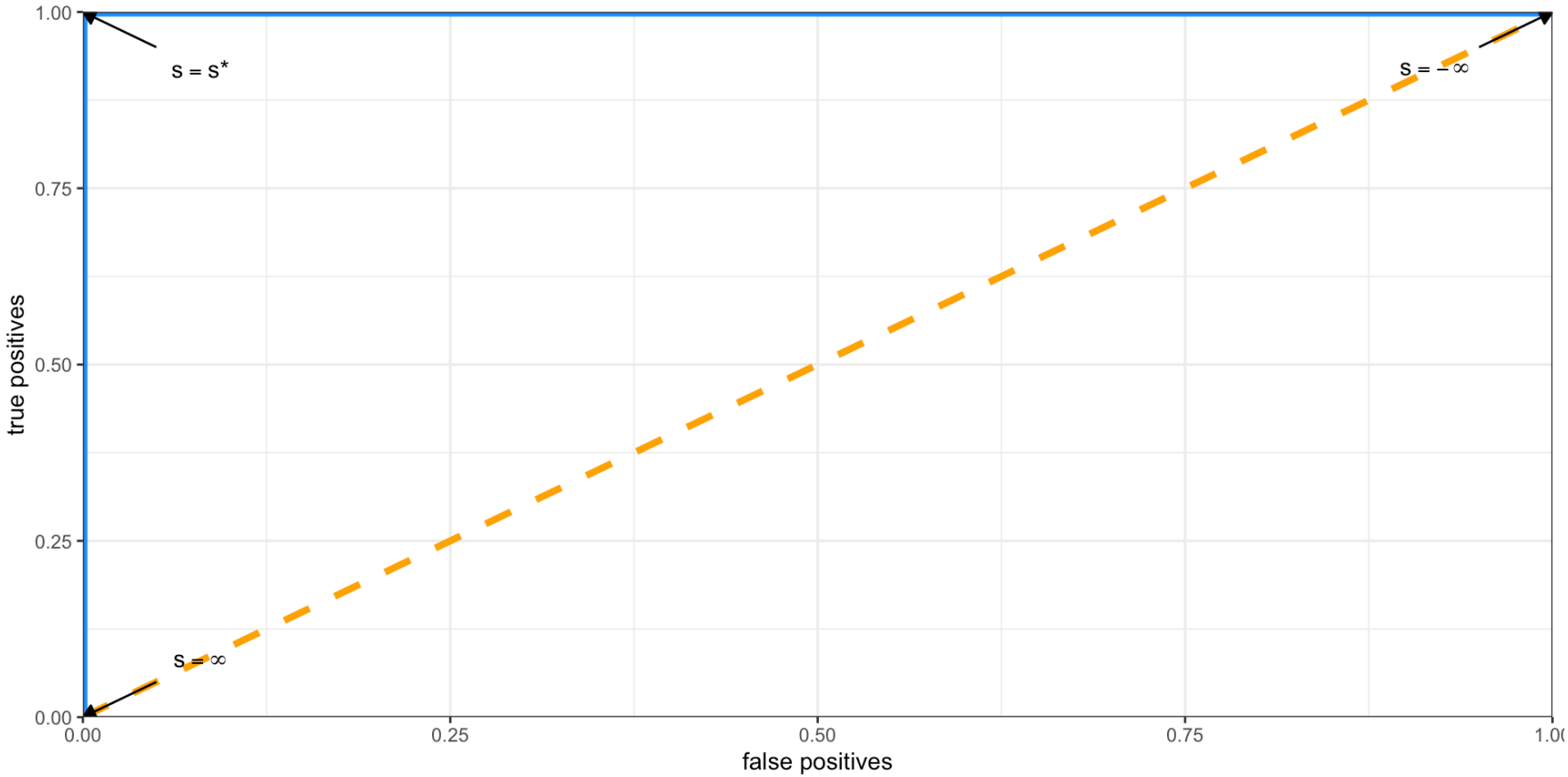


Figure 1: ROC curves of a perfect Score (solid, blue) and a random Score (dashed, orange)

The Logistic Regression model

The Logistic Regression model

Informal introductory example

We provide here some intuitions leading to the Logistic Regression model using a simulated data set from James et al. (2021) (the **Default** data set).

▾ Code

```
# Default data set (simulated) from ESLII/ISLR
default_data <- ISLR2::Default %>%
    as_tibble()

glimpse(default_data)
```

```
Rows: 10,000
Columns: 4
$ default <fct> No, No, No, No, No, No, No, No, No, No, No, No, No, No, No, No…
$ student <fct> No, Yes, No, No, No, Yes, No, Yes, No, No, Yes, Yes, No, No, N…
$ balance <dbl> 729.5265, 817.1804, 1073.5492, 529.2506, 785.6559, 919.5885, 8…
$ income  <dbl> 44361.625, 12106.135, 31767.139, 35704.494, 38463.496, 7491.55…
```

This is a toy data set used for teaching purposes containing information on ten thousand customers.

The aim here is to assess which customers will ***default*** on their credit card debt (the target or response variable) based on the current credit card ***balance*** and other individual characteristics (the predictors or feature vector).

The Logistic Regression model

Informal introductory example

We can start to explore the Default data with a scatterplot (Figure 2) of the target variable (***default***) with respect to a predictor (***balance***):

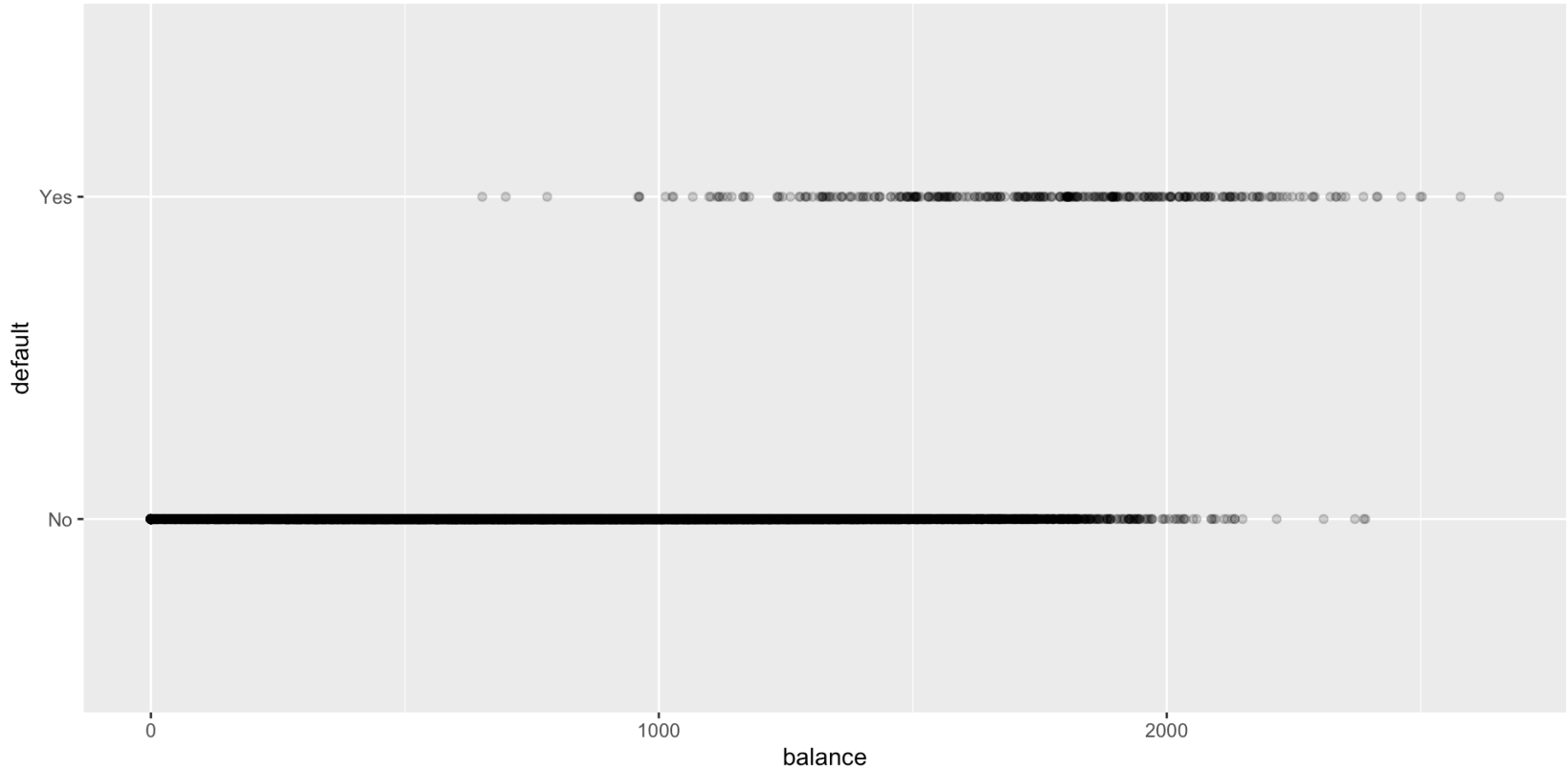
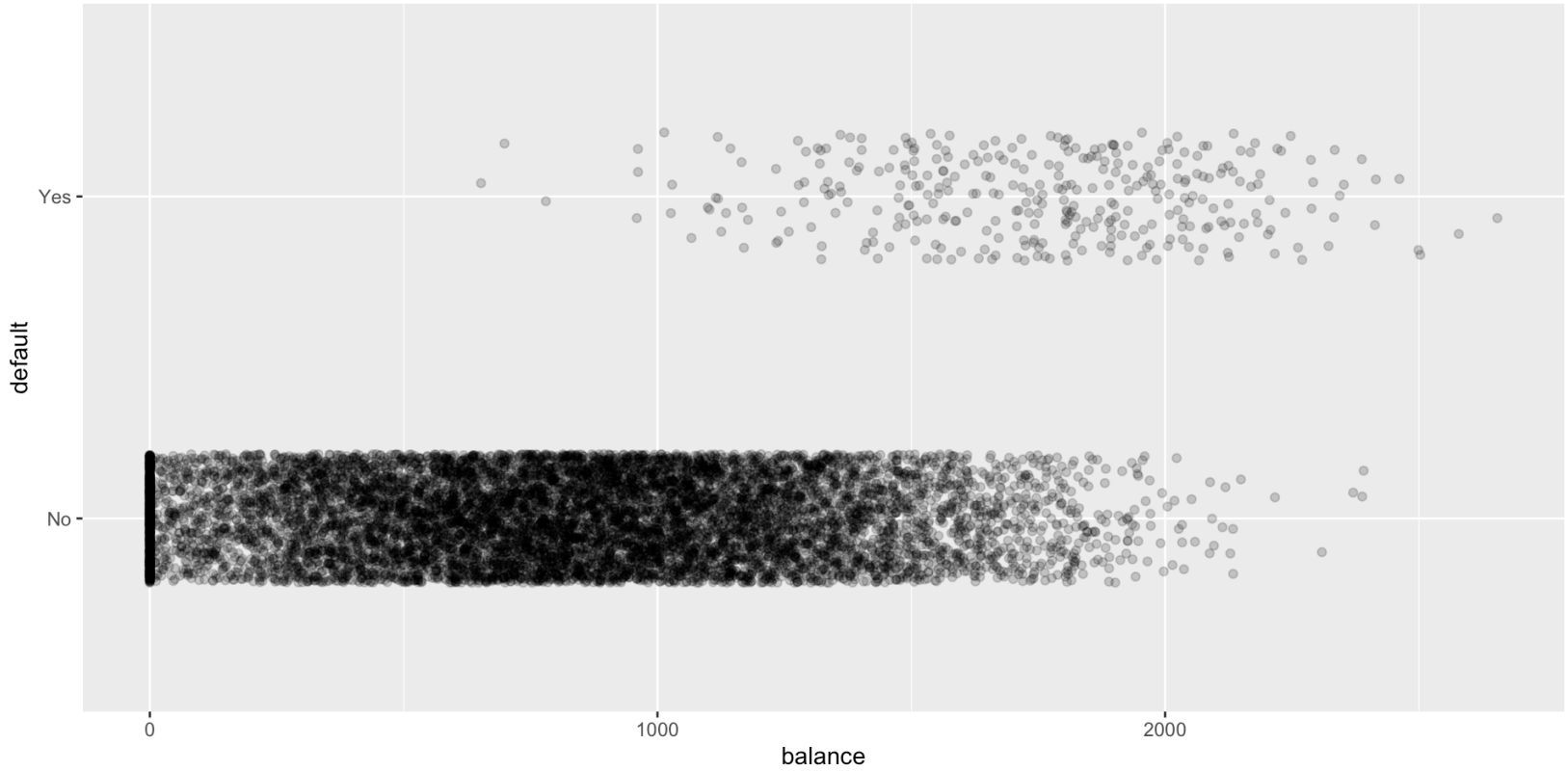


Figure 2: Scatterplot of variable ***default*** with respect to credit card ***balance*** for 10000 customers

The Logistic Regression model

Informal introductory example

In this scatterplot, all points fall on one of two parallel lines representing the absence (No) or occurrence (Yes) of ***default***. We "jitter" the data vertically to avoid overplotting. The plot below shows that the response variable is imbalanced towards the absence of default:



The Logistic Regression model

Informal introductory example

We also show the boxplots of credit cards ***balance*** with respect to ***default*** status:

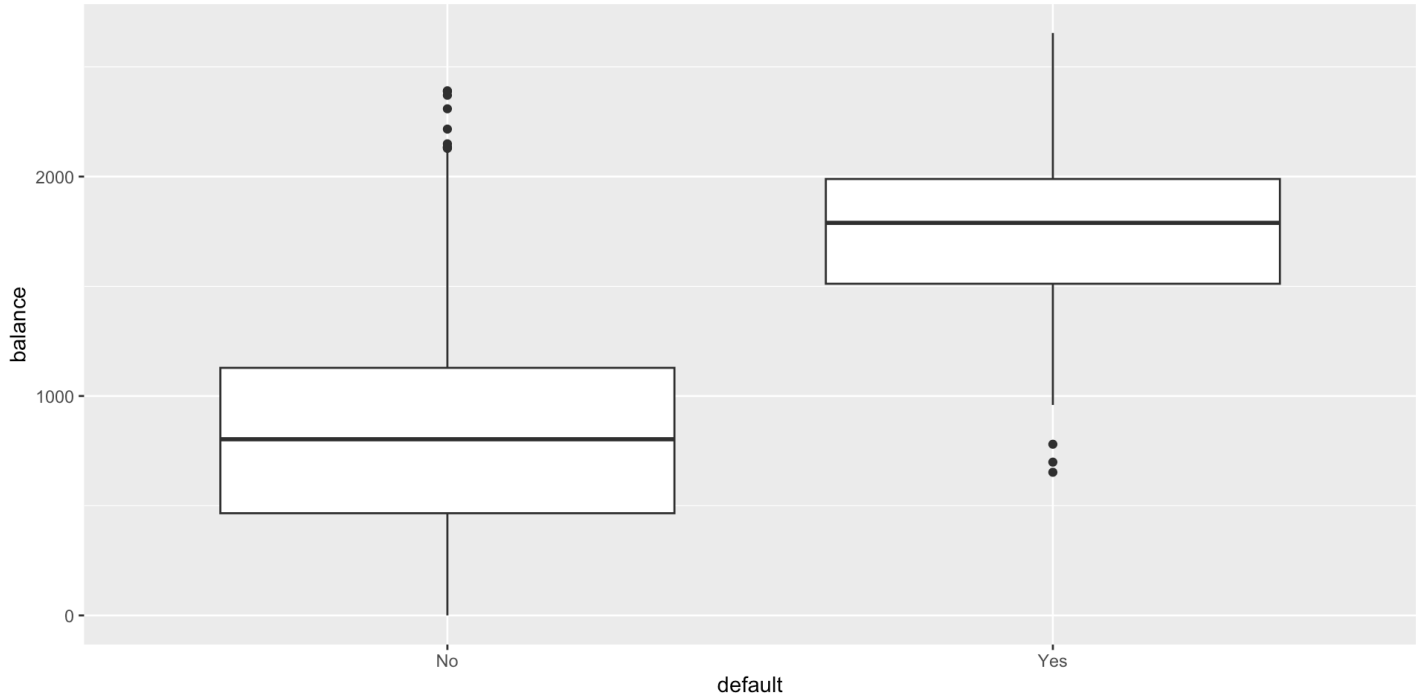


Figure 3: Variable ***balance*** with respect to ***default*** status

We can see from Figure 2 and Figure 3 that default tends to be more prevalent for accounts with a high balance. However it is difficult to guess a simple relationship between default and balance.

The Logistic Regression model

Informal introductory example

To investigate further we discretise the balance variables by classes of width 300 and compute the mean of response variable (***default*** is Yes) within each balance class:

```
# A tibble: 9 × 6
  balance_bins   min   max    No   Yes `Mean(default)`
  <fct>        <dbl> <dbl> <int> <int>           <dbl>
1 [0,300)          0   300  1497     0          0
2 [300,600)      300   600  1784     0          0
3 [600,900)      600   900  2305     3          0.0013
4 [900,1200)     900  1200  2098    19          0.009
5 [1200,1500)   1200  1500  1330    53          0.0383
6 [1500,1800)   1500  1800   527    96          0.154
7 [1800,2100)   1800  2100   115   114          0.498
8 [2100,2400)   2100  2400    11    41          0.788
9 [2400,2700)   2400  2700     0     7          1
```

The Logistic Regression model

Informal introductory example

Then we plot the mean of default (in red) within each balance class (of width $\mathbf{300\$}$):

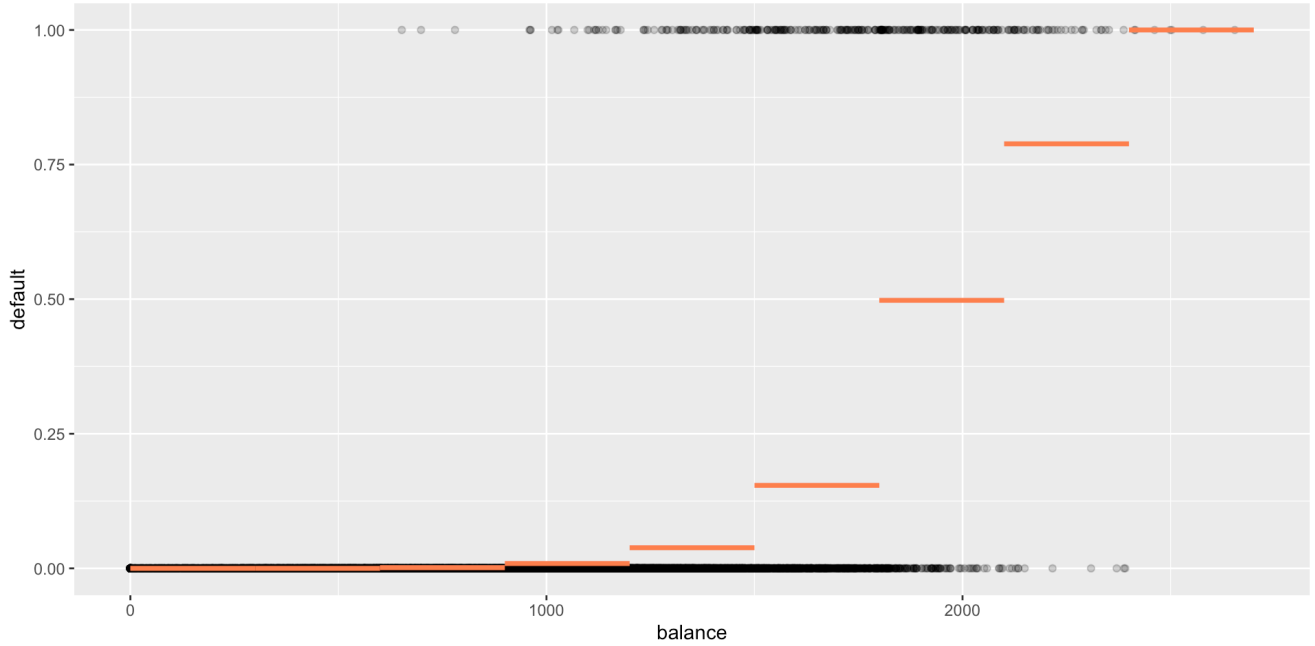


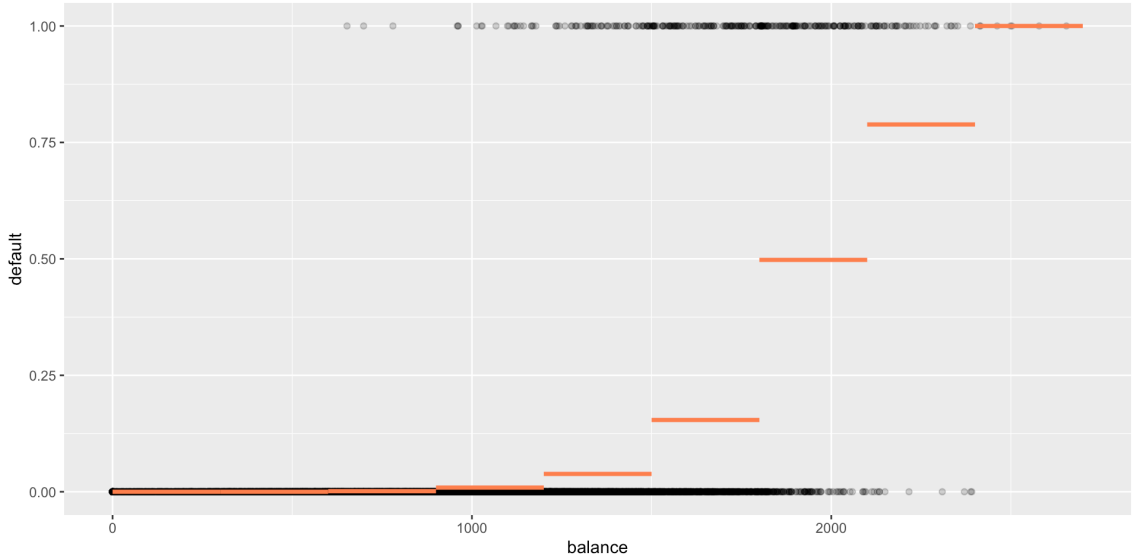
Figure 4: Mean occurrence of ***default*** within ***balance*** classes

The relationship between the mean occurrence of ***default*** and ***balance*** is easier to read.

Figure 4 clearly shows that as balance increases, the proportion of customers defaulting on their credit card increases.

The Logistic Regression model

Informal introductory example



We also notice that the mean default occurrence with respect to balance classes follows a kind of “S”-shaped curve or **sigmoid** function. Going further and informally, considering that the mean of default occurrence is an estimate of $\mathbf{E}[Y|X = x]$ for each balance classes an idea would be to model:

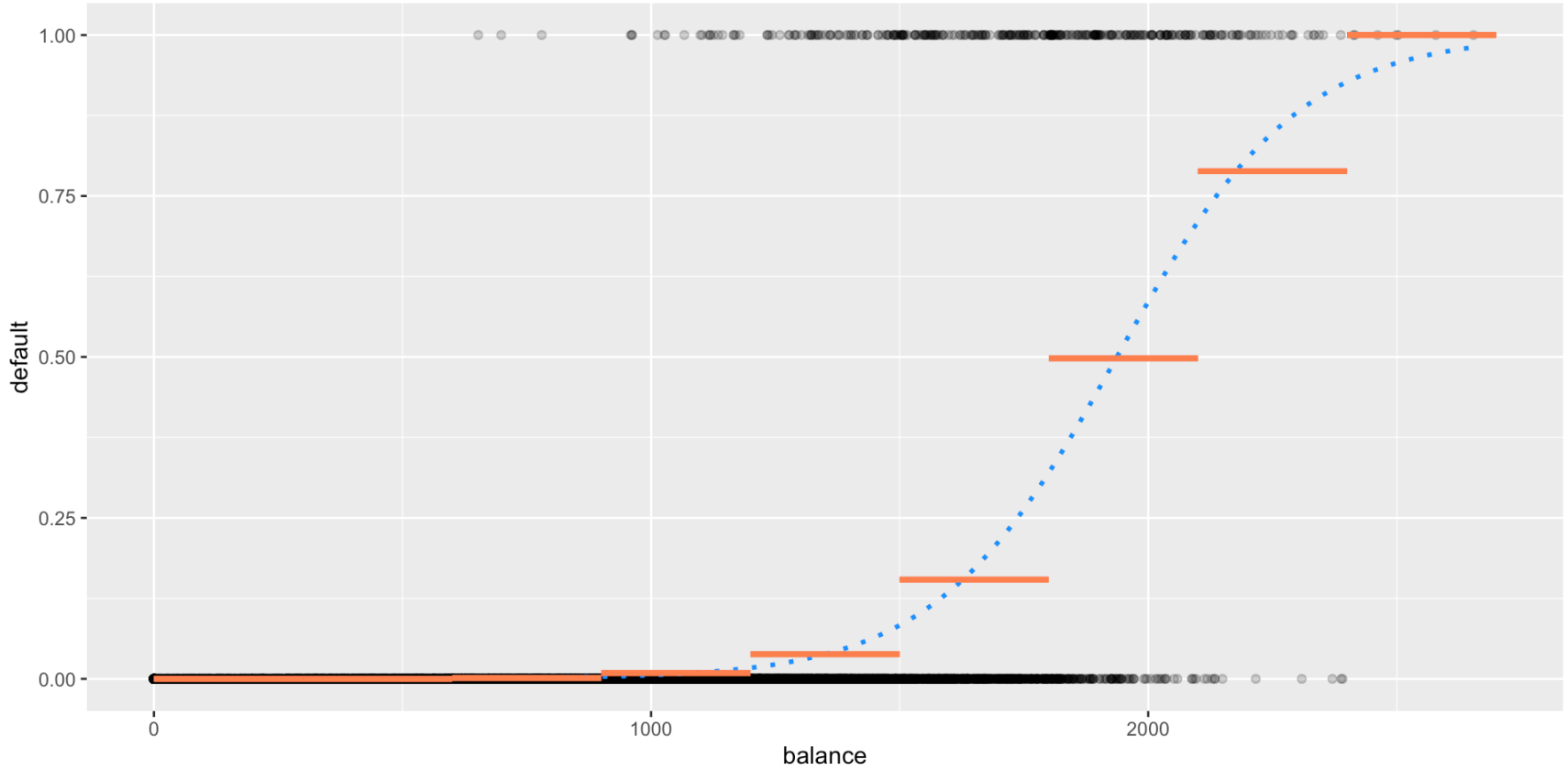
$$\mathbb{E}[Y|X = x] = \mu_\beta(x)$$

where μ_β is a **sigmoid** function in $[0, 1]$.

The Logistic Regression model

Informal introductory example

The Logistic Regression model uses the **sigmoid** function $\sigma : x \rightarrow \sigma(x) = \frac{e^x}{1+e^x}$ also known as the logistic function. Below a simple transform of this logistic function has been "fitted" (blue dots) to the `default ~ balance` data set:



The Logistic Regression model

A more formal definition - the discriminative approach

Reminding the statistical learning / scoring concepts introduced before. We try to predict the output `default` ($Y \in \{0, 1\}$) using a training set of inputs X: this is a binary classification problem.

We remind that to estimate an optimal classifier for output $Y \in \{0, 1\}$ using input $X = (X_1, \cdots, X_p)$ one approach was to:

- model the conditional distribution $Y|X$ (the **discriminative** approach),
- estimate $\eta(x)$ with:

$$\eta(x) = \mathbb{P}[Y = 1|X = x)] = \mathbb{E}[Y = 1|X = x)]$$

- $\eta(x)$ can be used as a Scoring function
- define the classifier f_s using a cutoff s and Scoring function $\eta(x)$ to predict output Y:

$$f_s(x) = \begin{cases} 1 & \text{if } \eta(x) \geq s \\ 0 & \text{otherwise} \end{cases}$$

The Logistic Regression model

A more formal definition

In the case of **Logistic Regression** classifier, we model:

$$Y|X = x \ \sim B(\eta(x))$$

with

$$\eta(x) = \sigma(x^T\beta) = \frac{exp(x^T\beta)}{1 + exp(x^T\beta)}$$

for some parameter $\beta = (\beta_1, \cdots, \beta_p) \in \mathbb{R}^p$, usually $x_1 = 1$ and β_1 is an intercept. σ is the sigmoid logistic function we have seen before.

In the literature is usual to denote $\eta(X) = p_\beta(X)$ or $\eta(X) = \pi_\beta(X)$.

From now, we will use the notation $p_\beta(X)$.

The Logistic Regression model

A more formal definition

Defining $\mathbf{logit} : x \to \log\left(\frac{x}{1-x}\right)$, which is the inverse of σ the logistic function (show it as an exercise), we have:

$$\mathrm{logit}(p_\beta(X)) = X^T\beta$$

The Logistic Regression model:

We are given $(x_i, y_i) \in \mathbb{R}^p \times \{0, 1\}$, $i = 1, \cdots, n$

The Logistic Regression model assumes that outputs y_i are independent Bernoulli with parameter $p_\beta(x_i)$ depending on x_i:

$$\mathrm{logit}(p_\beta(x_i)) = x_i^T\beta$$

The Logistic Regression model

How to fit with R

The syntax to fit the Logistic model in R using `glm()` is:

$$\texttt{glm(y} \sim \texttt{x, data} = \text{dataframe, family = binomial(link = ''logit'')}$$

The formula $\mathbf{y} \sim \mathbf{x}$ depicts the model (i.e. inputs are X, output is Y) and the `data=` argument points to the training set contained in a R dataframe (or tibble). This is quite similar to the `lm()` function.

We also need to specify the distribution for the conditional Y values (binomial) and the link function (logit) via the `family=` argument.

The Logistic Regression model

How to fit with R

For our default example:

The command `summary` produces result summaries of the fitted model:

```
Call:
glm(formula = default ~ ., family = "binomial", data = default_data)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **
balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
income       3.033e-06  8.203e-06   0.370  0.71152
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2920.6  on 9999  degrees of freedom
Residual deviance: 1571.5  on 9996  degrees of freedom
AIC: 1579.5

Number of Fisher Scoring iterations: 8
```

We will see in the next lesson, how this model is fitted in practice and how to interpret or understand what is printed by the `summary` function.

References

Cornillon, P.-A., Matzner-Løber, E., Hengartner, N., & Rouvière, L. (2019). *Régression avec r - 2e édition*. EDP Sciences.

Desbois, D. (2008). Introduction to scoring methods: financial problems of farm holdings. *Case Studies in Business, Industry and Government Statistics*, *2*(1), 56–76. https://hal.science/hal-01172847

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*.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in r*. Springer US. https://doi.org/10.1007/978-1-0716-1418

Thomas, L., Edelman, D., & Crook, J. (2002). Credit scoring and its applications. In *Monographs on Mathematical Modeling & Computation*.