Exam - In class

Table of contents

1 Introduction

The goal of this exam is to build and assess a predictive model to quantify the probability of corporate default **with a 1-year horizon**, using annual financial statement data reported by companies. Financial statement data (both financial statement items and ratios) relating to these companies is available in a data set.

The data set is anonymized, neither companies names/identifiers nor fiscal years are given.

When a corporate default (column `Y`) occurs for a company during a given fiscal year, say `N`, the observation (ie row in data set) of the preceding fiscal year (`N-1`) for this company shows a value of `1` in the column `Y`, `0` otherwise.

Your goal is to predict corporate default with a 1-year horizon.

We briefly describe the columns of the data set (Rows: 4,353 Columns: 44):

- `Y`: 0 for a healthy company (ie no default during the next fiscal year), 1 for a company in default during the next fiscal year;

First a set of financial statements items mainly coming from companies income statements or balance sheets:

- `capx`: Capital Expenditures (see here).
- `che`: Cash and Equivalents (see here).
- `cogs`: Cost of Goods Sold (see here).
- `dp`: Depreciation and Amortization (see here).
- `act`: Current Assets (see here.
- `lct`: Current Liabilities (see here).
- `ebit`: Earnings Before Interest and Taxes (see here).
- `ebitda`: Earnings Before Interest, Taxes, Depreciation and Amortization (see here).
- `re`: Retained Earnings (see here).
- `gp`: Gross Profit (see here).
- `ib`: (Net) Income Before Extraordinary Items (see here).
- `xint`: Interest Expense (see here).
- `dltt`: Long-Term Debt (see here).
- `ni`: Net Income (see here or here).
- `ppent`: Property, Plant, and Equipment (see here).
- `sale`: Net Sales (see here).
- `seq`: (Total Parent) Stockholders' Equity (see here).
- `at`: Total Assets (see here).
- `lt`: Total Liabilities (see here).
- `invt`: Inventories (see here).
- `rect`: (Accounts) Receivables (see here).
- `wcap`: Working Capital (see here).
- `mktval`: Market Value of Equity (see here).

Then a set of financial statements ratios, loosely inspired from those of the Desbois case study presented in class (r1 to r37, we keep Desbois presentation):

Capitalization ratios:

- `r1`: Total Liabilities / Total Assets
- `r2`: Stockholders' Equity / Invested Capital
- `r3`: Current Liabilities / Total Liabilities
- `r4`: Current Liabilities / Total Assets
- `r5`: (Total Liabilities - Current Liabilities) / Total Assets

Weight of the debt:

- `r6`: Total Liabilities / Gross Profit
- `r7`: (Total Liabilities - Current Liabilities) / Gross Profit
- `r8`: Current Liabilities / Gross Profit

Liquidity:

- `r11`: Working Capital / Gross Profit
- `r12`: Working Capital / (Cost of Goods Sold - Interest Expense)
- `r14`: Current Liabilities / Current Assets

Debt servicing:

- `r17`: Interest Expense / Total Liabilities
- `r18`: Interest Expense / Gross Profit
- `r21`: Interest Expense / EBITDA

Capital profitability:

- `r24`: EBITA / Total Assets

Earnings:

- `r28`: EBITA / Gross Profit
- `r30`: Net Income / Gross Profit
- `r32`: (EBITA - Interest Expense) / Gross Profit

Productive activity:

- r36: (Total Assets - Current Assets) / Gross Profit
- r37: Gross Profit / Total Assets

We load the data set and briefly inspect it:

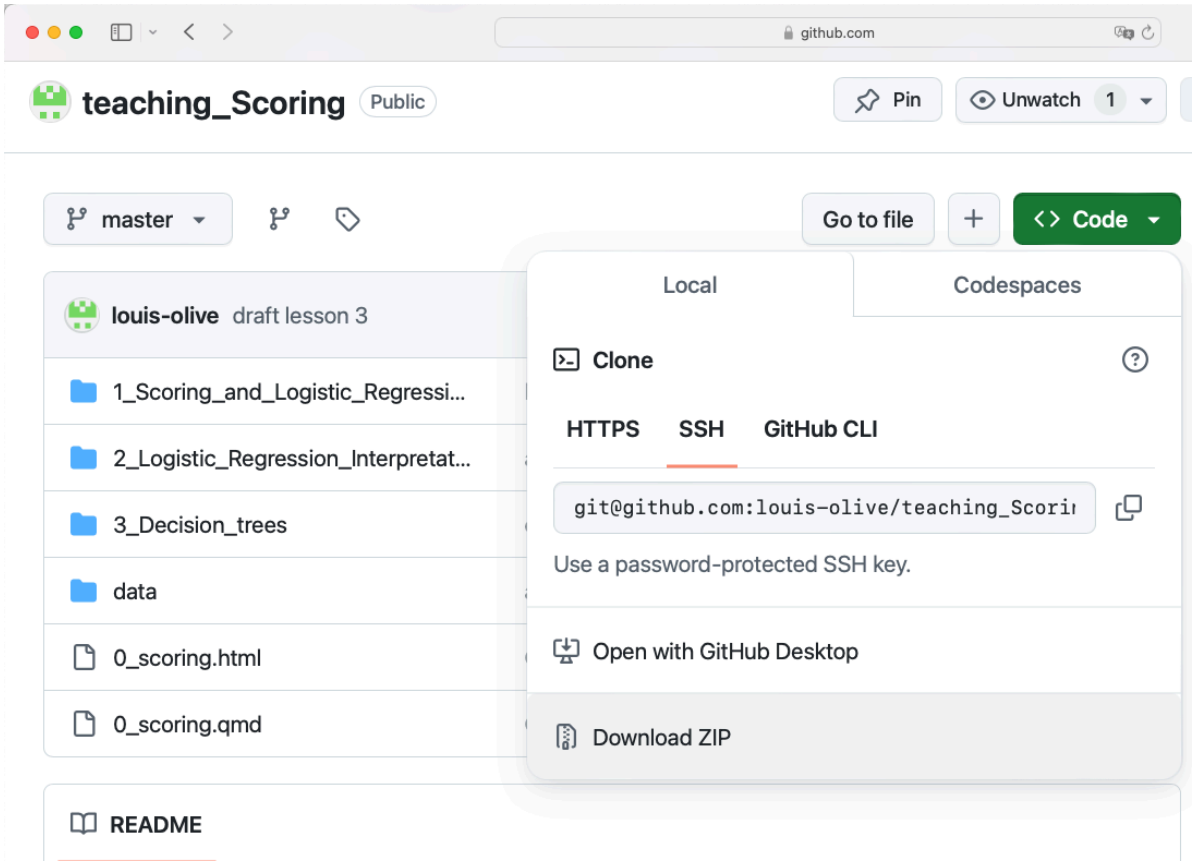
```
# loading data and inspecting it:
data_fin_exam <- readRDS('data/data_fin_exam.rds')
glimpse(data_fin_exam)
```

```
Rows: 4,353
Columns: 44
$ Y      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ capx   <dbl> 2.301, 60.223, 4.346, 42.619, 32.169, 0.210, 31.845, 0.275, 952~
$ che    <dbl> 3.132, 12.678, 5.082, 58.623, 63.305, 1.482, 18.014, 1.454, 378~
$ cogs   <dbl> 11.378, 332.119, 385.265, 729.995, 1476.918, 8.116, 180.126, 3.~
$ dp     <dbl> 2.040, 53.072, 4.418, 49.681, 28.066, 0.385, 18.260, 0.191, 484~
$ act    <dbl> 6.351, 181.599, 128.932, 692.991, 597.103, 5.882, 171.237, 5.63~
$ lct    <dbl> 2.976, 128.618, 58.159, 358.271, 335.409, 5.233, 65.142, 1.006,~
$ ebit   <dbl> 2.061, 76.904, 29.493, 105.547, 79.014, 0.475, 32.654, 1.383, 1~
$ ebitda <dbl> 4.101, 129.976, 33.911, 155.228, 107.080, 0.860, 50.914, 1.574,~
$ re     <dbl> 6.873, -106.904, 50.140, 72.500, -31.142, -2.850, 55.968, 3.841~
$ gp     <dbl> 9.212, 174.608, 111.464, 567.290, 349.877, 6.415, 91.762, 5.138~
$ ib     <dbl> 1.789, 0.125, 14.158, 19.444, 44.710, -0.054, 15.811, 0.851, 13~
$ xint   <dbl> 0.003, 77.560, 3.446, 63.775, 20.069, 0.131, 9.980, 0.036, 54.2~
$ dltt   <dbl> 0.000, 867.944, 46.700, 782.249, 433.135, 1.737, 140.175, 0.203~
$ ni     <dbl> 1.789, 0.125, 14.158, 13.833, 39.811, -0.054, 15.811, 0.851, 13~
$ ppent  <dbl> 12.150, 554.296, 9.861, 481.554, 232.132, 1.947, 195.958, 0.774~
$ sale   <dbl> 20.590, 506.727, 496.729, 1297.285, 1826.795, 14.531, 271.888, ~
$ seq    <dbl> 7.572, 82.717, 106.510, 1131.989, 464.633, 6.742, 186.312, 5.32~
$ at     <dbl> 20.010, 1230.512, 222.119, 2329.268, 1311.587, 13.961, 420.046,~
$ lt     <dbl> 12.438, 1147.795, 115.609, 1197.279, 846.954, 6.970, 233.734, 1~
$ invt   <dbl> 0.115, 77.086, 43.631, 309.277, 229.805, 1.007, 77.577, 1.300, ~
$ rect   <dbl> 3.010, 84.345, 74.763, 258.471, 272.424, 2.979, 60.943, 2.808, ~
$ wcap   <dbl> 3.375, 52.981, 70.773, 334.720, 261.694, 0.649, 106.095, 4.633,~
$ mktval <dbl> 14.784000, 0.326250, 217.974375, 1045.883400, 682.119000, 24.40~
$ r1     <dbl> 0.6215892, 0.9327784, 0.5204823, 0.5140151, 0.6457475, 0.499247~
$ r2     <dbl> 1.00000000, 0.08700999, 0.69518961, 0.59135228, 0.51754239, 0.7~
$ r3     <dbl> 0.2392668, 0.1120566, 0.5030664, 0.2992377, 0.3960180, 0.750789~
$ r4     <dbl> 0.1487256, 0.1045240, 0.2618371, 0.1538127, 0.2557276, 0.374829~
$ r5     <dbl> 0.472863568, 0.828254418, 0.258645141, 0.360202433, 0.390019877~
$ r6     <dbl> 1.3501954, 6.5735533, 1.0371869, 2.1105237, 2.4207193, 1.086516~
$ r7     <dbl> 1.02713851, 5.83694332, 0.51541305, 1.47897548, 1.46207096, 0.2~
```

```
$ r8     <dbl> 0.3230569, 0.7366100, 0.5217738, 0.6315482, 0.9586483, 0.815744~
$ r11    <dbl> 0.36636995, 0.30342825, 0.63494043, 0.59003332, 0.74796000, 0.1~
$ r12    <dbl> 0.20471916, 0.17708086, 0.15406468, 0.31041972, 0.15396971, 0.0~
$ r14    <dbl> 0.4685876, 0.7082528, 0.4510827, 0.5169923, 0.5617272, 0.889663~
$ r17    <dbl> 0.0002411963, 0.0675730422, 0.0298073679, 0.0532666154, 0.02369~
$ r18    <dbl> 0.0003256622, 0.4441949968, 0.0309158114, 0.1124204551, 0.05736~
$ r21    <dbl> 0.0007315289, 0.5967255493, 0.1016189437, 0.4108472698, 0.18742~
$ r24    <dbl> 0.20494753, 0.10562758, 0.15267042, 0.06664240, 0.08164155, 0.0~
$ r28    <dbl> 0.4451802, 0.7443874, 0.3042328, 0.2736308, 0.3060504, 0.134060~
$ r30    <dbl> 0.1942032132, 0.0007158893, 0.1270185890, 0.0243843537, 0.11378~
$ r32    <dbl> 0.44485454, 0.30019243, 0.27331695, 0.16121032, 0.24869025, 0.1~
$ r36    <dbl> 1.4827399, 6.0072448, 0.8360278, 2.8843748, 2.0421005, 1.259392~
$ r37    <dbl> 0.4603698, 0.1418987, 0.5018211, 0.2435486, 0.2667585, 0.459494~
```

2 Instructions

- The exam is open documents, open browser. If copying large chunks of codes from the browser, give a reference (link to website, stackoverflow, stats.stackexchange, copy ChatGPT Q/A in appendix etc). You can reuse code from the two first lessons hosted here: https://github.com/louis-olive/teaching_Scoring/, click on Code/Download ZIP for the last version:



- The first parts of the exam are to be performed in-class (**TO DO in-class** in the exam document). The in-class exam will last two hours (10:30-12:30).
- You must use the R programming language, preferably through RStudio. Your code for analysis should use one of the following formats: preferably quarto Markdown (.qmd, as done in the course), but you might prefer R Markdown (.Rmd) or an R file (.R).
- Packages that you may need besides `base R` and `stats` (`glm()`, `step()`) that have been used in the course include:

`tidyverse`, `broom`, `class`, `ROCR`, `car`, `aod`, `rsample`, `bestglm`, `glmnet`, `glmnetUtils`, `splines`, `rpart`, `rpart.plot`, `ada`, `gbm`, `xgboost`

You can install them using the following code (uncomment):

```
# # UNCOMMENT IF NEEDED
# # https://statsandr.com/blog/an-efficient-way-to-install-and-load-r-packages/
# # Package names
# packages <- c("tidyverse", "ROCR", "car", "aod", "broom", "rsample", "bestglm",
```

```
# "glmnet", "glmnetUtils", "splines")
#
# # Install packages not yet installed
# installed_packages <- packages %in% rownames(installed.packages())
# if (any(installed_packages == FALSE)) {
#   install.packages(packages[!installed_packages])
# }
#
# # Packages loading
# invisible(lapply(packages, library, character.only = TRUE))
#
#
# # Additional packages used throughout the course but not needed for the analysis
# additional_packages <- c( "purrr", "pROC", "foreign", "patchwork", "class",
# "scales", "rpart", "rpart.plot", "DescTools")
```

Check that it works on your computer before the course.

- For your report: either render a .html file of your analysis (when using .qmd/.Rmd) or provide a rich document with your text plus tables/plots of your analysis (.docx or .pdf). In case you don't want to work with markdown/notebooks, no recommendation but LibreOffice is free and runs cross-platforms.
- The code file and report's general readability especially for the take home will impact the grading as well as the richness of their content (do not hesitate to comment on your intents, assumptions, findings, conclusions, especially in the take home part).
- The .html file for report should be readable in a standard web browser (Chrome, Safari, Firefox, Edge …), alternatively .docx/.pdf should be valid. The .qmd/.Rmd/.R code file should run without errors (if something is not working as you wish, comment in the code with your intents). They should be posted before 12:30 on 2 October 2024 for the in-class part to my two email addresses `louis.olive@ut-capitole.fr`, `louis.olive@gmail.com` (in case the first one encounters issues) with subject `SCORING EXAM - YOUR NAME`. You can prepare your email in advance to save time at the end of exam.
- Allow yourself at least 15 minutes before the end to check your .qmd/.Rmd/.R file is running and you have a readable report or code. If you finish before the end, and are happy with the result post me your code/report and take a well deserved rest!
- If you are not happy with some or all parts of the in-class analysis, you might complete/correct/improve it at home, if it improves your grading a maximum of half of points will be given (for each relevant improvement).
- For the take-home part the deadline is 9 October 2024 08:00.

- Regarding the grading:
  - 50% of the total points for the in-class part, 50% for the at-home part
  - in-class data analysis (40%): 15% / 5% / 10% / 10% of the total points for each parts: "Desbois" ratios, "Altman" ratios, Lasso on financial statements items and Models assessment on holdout dataset
  - in-class extra exercise (10%): simulating data and assessing boundary decisions

You should follow the following plan in you report:

3 "Desbois" ratios (15% total points)

TO DO in-class

- First explore the data set using variables `Y` and 'r1:r37", for example:
  - providing descriptive statistics,
  - showing correlated features,
  - showing/plotting individual features "interaction" with the response variable

  Visualizations are expected.

  After this step, you can remove observations (rows) from data if justified.

  You might also need to remove some feature variables is perfectly co-linear or any other reason.
- Secondly fit a "full" logistic regression model (**full_model_desbois**) on the data set using variables `Y` and kept features from exploration step among `r1:r37`.
- Then use stepwise logistic regression (forward or backward, using the penalization/criterion of your choice) (**stepwise_model_desbois**) on the data set using variables `Y` and kept features from exploration step among `r1:r37`.
- Compare the **full_model_desbois** and **stepwise_model_desbois** using a Likelihood Ratio Test (LRT), ie test if **full_model_desbois** fits significantly better than the **stepwise_model_desbois**.
- Compare predicted probabilities to observed probabilities for **stepwise_model_desbois** using either Hosmer & Lemeshow test or a Calibration Plot.

4 "Altman" ratios (5% total points)

TO DO in-class

- First, using the data set at hand, create new predictors closest as possible as Altman's Z-Score components (X1-X5) as shown below:

ent ratio-profiles. The final discriminant function is as follows:

$$\text{(I)} \quad Z = .012X_1 + .014X_2 + .033X_3 + .006X_4 + .999X_5$$

where X_1 = Working capital/Total assets
X_2 = Retained Earnings/Total assets
X_3 = Earnings before interest and taxes/Total assets
X_4 = Market value equity/Book value of total debt
X_5 = Sales/Total assets
Z = Overall Index

- Secondly, fit a logistic regression model **model_altman** using only these predictors, then:
  - give an interpretation for the coefficient `X3 = EBIT / Total Assets`
  - assess the "significance" of `X3 = EBIT / Total Assets` coefficient
  - give a confidence interval for `X3 = EBIT / Total Assets`

5 Financial items / Lasso (10% total points)

TO DO in-class

- First explore the data set using variables `Y` and `capx:mktval`, for example:
  - providing descriptive statistics,
  - showing correlated features,
  - showing/plotting individual features "interaction" with the response variable

  Visualizations are expected.

  After this step, you can remove observations (rows) from data if justified.

  You might also need to remove some feature variables is perfectly co-linear or any other reason.
- Secondly fit a "full" logistic regression model (**full_model_items**) on the data set using variables `Y` and kept features from last step among `capx:mktval`.

- Then using function `glmnet::cv.glmnet` or more conveniently `glmnetUtils::cv.glmnet` as shown in lesson 2 (for example in the cross-validation function defined at the end of the lesson): select a "best value" for the lasso parameter "lambda" giving a penalized model **lasso_model_items**.

Functions `glmnet::cv.glmnet` / `glmnetUtils::cv.glmnet` fit the lasso path (ie multiple penalized models for multiple values of lambda) with the selection of the best lambda by K-Fold Cross-Validation (by default 10-fold) using a given criterion such as the AUC (`type.measure = "auc"`).

The function computes two optimal values: `lambda.min` is the value of lambda that gives minimum mean Cross-validated criterion; `lambda.1se` is the value of lambda that gives the most regularized model (highest lambda) such that the Cross-validated criterion for this lambda is within one standard error of the minimum, it favours penalization/parsimony versus `lambda.min` (see here).

(example usage `model_cv_glmnet <- glmnetUtils::cv.glmnet(Y ~ ., data= YOUR_DATA, family="binomial", alpha=1, type.measure = "auc")`).

By default the `predict` function for a `cv.glmnet` model uses `lambda.1se`, but we can specify any value of lambda, in particular `lambda.min`, also note that the predict function outputs a `matrix` (`predict` functions in `R` outputs a vector in general) that might need to be converted to a vector (for latter use with `ROCR` package).

(example usage `predicted_proba <- as.vector(predict(model_cv_glmnet, newdata = YOUR_(NEW)_DATA, s = model_cv_glmnet$lambda.1se, type = "response")`)

6 Models assessment (10% total points)

TO DO in-class

You are given a holdout/testing data set to assess the preceding models.

- First (if you have already completed the task) create the Altman's predictors for this data set.
- Then plot the ROC Curves and compare the AUC of **full_model_desbois**, **stepwise_model_desbois**, **model_altman**, **full_model_items**, **lasso_model_items**

We load the holdout/testing data set and briefly inspect it:

```
# loading data and inspecting it:
data_fin_holdout <- readRDS('data/data_fin_holdout.rds')
glimpse(data_fin_holdout)
```

```
Rows: 461
Columns: 44
$ Y      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ capx   <dbl> 960.000, 68.097, 882.000, 94.900, 598.000, 681.000, 2928.000, 3~
$ che    <dbl> 1236.000, 25.539, 5926.000, 532.700, 1806.000, 1832.000, 7388.0~
$ cogs   <dbl> 6193.000, 377.460, 57762.000, 2164.800, 9203.000, 5521.000, 261~
$ dp     <dbl> 398.000, 55.714, 900.000, 102.500, 678.000, 773.000, 3033.000, ~
$ act    <dbl> 1787.000, 199.987, 21045.000, 1762.000, 4884.000, 5919.000, 319~
$ lct    <dbl> 2942.000, 75.301, 13173.000, 1496.500, 5577.000, 2836.000, 2613~
$ ebit   <dbl> 775.000, 46.263, 1663.000, 246.600, 1176.000, 1767.000, 2750.00~
$ ebitda <dbl> 1173.000, 101.977, 2563.000, 349.100, 1854.000, 2540.000, 5783.~
$ re     <dbl> 4086.000, 685.476, 14846.000, 1203.900, 4893.000, 11202.000, 25~
$ gp     <dbl> 2071.000, 130.741, 4584.000, 3238.100, 2271.000, 5606.000, 1242~
$ ib     <dbl> 437.000, 50.943, 1279.000, -19.500, 566.000, 1630.000, -67.000,~
$ xint   <dbl> 91.000, 18.871, 313.000, 134.600, 317.000, 105.000, 1108.000, 2~
$ dltt   <dbl> 1617.000, 258.458, 6504.000, 1581.600, 6518.000, 3473.000, 2281~
$ ni     <dbl> 437.000, 50.943, 1279.000, -19.500, 566.000, 1624.000, -67.000,~
$ ppent  <dbl> 6781.000, 726.251, 9758.000, 570.200, 4709.000, 4542.000, 15322~
$ sale   <dbl> 8264.000, 508.201, 62346.000, 5402.900, 11474.000, 11127.000, 3~
$ seq    <dbl> 3751.000, 702.555, 17173.000, -412.400, 2949.000, 7794.000, 131~
$ at     <dbl> 10912.000, 1111.893, 39769.000, 3010.000, 17360.000, 15641.000,~
$ lt     <dbl> 7161.000, 409.338, 22564.000, 3414.700, 14341.000, 7825.000, 61~
$ invt   <dbl> 60.000, 8.187, 8831.000, 542.000, 1274.000, 1653.000, 8614.000,~
$ rect   <dbl> 366.000, 160.289, 4405.000, 418.900, 1631.000, 1812.000, 14503.~
$ wcap   <dbl> -1155.000, 124.686, 7872.000, 265.500, -693.000, 3083.000, 5835~
$ mktval <dbl> 7496.35490, 487.19096, 26157.45000, 672.44800, 20994.14545, 337~
$ r1     <dbl> 0.6562500, 0.3681451, 0.5673766, 1.1344518, 0.8260945, 0.500287~
$ r2     <dbl> 0.6987705, 0.7310567, 0.7243241, -0.3504121, 0.3092167, 0.69040~
$ r3     <dbl> 0.41083648, 0.18395800, 0.58380606, 0.43825226, 0.38888501, 0.3~
$ r4     <dbl> 0.26961144, 0.06772324, 0.33123790, 0.49717608, 0.32125576, 0.1~
$ r5     <dbl> 0.38663856, 0.30042189, 0.23613870, 0.63727575, 0.50483871, 0.3~
$ r6     <dbl> 3.4577499, 3.1309077, 4.9223386, 1.0545382, 6.3148393, 1.395825~
$ r7     <dbl> 2.03718011, 2.55495216, 2.04864747, 0.59238442, 3.85909291, 0.8~
$ r8     <dbl> 1.4205698, 0.5759555, 2.8736911, 0.4621537, 2.4557464, 0.505886~
$ r11    <dbl> -0.55770159, 0.95368706, 1.71727749, 0.08199253, -0.30515192, 0~
$ r12    <dbl> -0.16500000, 0.32189243, 0.13236926, 0.05397219, -0.07449210, 0~
$ r14    <dbl> 1.6463346, 0.3765295, 0.6259444, 0.8493190, 1.1418919, 0.479135~
$ r17    <dbl> 0.012707722, 0.046101266, 0.013871654, 0.039417811, 0.022104456~
$ r18    <dbl> 0.043940126, 0.144338807, 0.068280977, 0.041567586, 0.139586085~
$ r21    <dbl> 0.07757886, 0.18505153, 0.12212251, 0.38556288, 0.17098166, 0.0~
$ r24    <dbl> 0.10749633, 0.09171476, 0.06444718, 0.11598007, 0.10679724, 0.1~
$ r28    <dbl> 0.5663930, 0.7799925, 0.5591187, 0.1078101, 0.8163804, 0.453086~
$ r30    <dbl> 0.211009174, 0.389648236, 0.279013962, -0.006022050, 0.24922941~
```

```
$ r32    <dbl> 0.52245292, 0.63565370, 0.49083770, 0.06624255, 0.67679436, 0.4~
$ r36    <dbl> 4.4060840, 6.9749046, 4.0846422, 0.3854112, 5.4936151, 1.734213~
$ r37    <dbl> 0.1897911, 0.1175842, 0.1152657, 1.0757807, 0.1308180, 0.358417~
```

7 Exercise: Simulation and boundary decision (10% total points)

TO DO in-class

Write a function simulating the following data set:

- Class 0: mixture (ie two buckets a,b chosen randomly with probability $\frac{1}{2}$) of Gaussian $\mu_{0a} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$ or $\mu_{0b} = \begin{bmatrix} 1 \\ -4 \end{bmatrix}$ and $\Sigma_{0a} = \Sigma_{0b} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$
- Class 1: Gaussian with $\mu_1 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$ and $\Sigma_1 = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$

Simulate a training test of 200 observations (100 for each Class 0/1).

Simulate a testing test of 2000 observations (1000 for each Class 0/1).

Fit a logistic regression (`Y ~ x1 + x2`) on the training set (**model1**).

Then a logistic regression model with interactions (`Y ~ x1 + x2 + I(x1*x2) + I(x1^2) + I(x2^2)`) (**model2**).

Assess the misclassification error on the testing set.

Bonus **TO DO at home** (5% total points): derive the expression of the Bayes boundary decision, show it on a bivariate (x1, x2) plot together with training set and **model1**/**model2** boundary decisions.