TD 2 — Régression logistique (R)

Année universitaire 2025–2026 ▪ Parcours Économie de la santé & Développement durable

Pierre Beaucoral

Ce TD reprend l'exemple et le dictionnaire de variables (`age`, `ed`, `employ`, `address`, `income`, `debtinc`, `credebt`, `othdebt`, `default`) décrits dans la ressource d'origine.
Données à récupérer: **`bankloanT.xls`** (ENT).

Packages utilisés : `tidyverse`, `readxl`, `janitor`, `broom`, `ResourceSelection`, `pROC`, `gt`, `ggplot2`.

Rappel de cours : Modèles Logit et Probit

Quand utiliser ces modèles ?

La **régression logistique** et la **régression probit** sont utilisées lorsque la variable à expliquer est **binaire** :

$Y_i \in 0, 1$

- Exemple : défaut de paiement / pas de défaut
- Objectif : estimer la **probabilité** $P(Y_i = 1 \mid X_i)$ en fonction de caractéristiques X_i

Les variables explicatives peuvent être **quantitatives ou qualitatives**.

Pourquoi ne pas utiliser une régression linéaire ?

Une régression linéaire classique :

$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$

peut donner des valeurs **prédites hors de [0, 1]**, ce qui est absurde pour une probabilité.

Il faut donc introduire une **fonction de lien** qui transforme l'intervalle (0, 1) en $\mathbb{R}$.

Fonction de lien du modèle Logit

$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) \quad \text{et} \quad p = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$

Le modèle s'écrit :

$\text{logit}(P(Y_i = 1)) = \beta_0 + \beta 1 Xi1 + \cdots + \beta_p Xip$

Les coefficients β_j s'interprètent via les **odds-ratios** : $OR_j = e^{\beta_j}$

Fonction de lien du modèle Probit

Le modèle Probit suppose qu'il existe une **variable latente** (Y_i^*) telle que :

$Y_i^* = \beta_0 + \beta 1 Xi1 + \cdots + \beta_p Xip + \varepsilon_i \quad \text{avec} \quad \varepsilon_i \sim \mathcal{N}(0,1)$

et on observe :

$Y_i = 1$ si $Y_i^* > 0$

d'où :

$P(Y_i = 1) = \Phi(\beta_0 + \beta_1 X_i)$ où Φ est la fonction de répartition de la loi normale centrée réduite.

Comparaison graphique des liens Logit et Probit

```r
library(ggplot2)

x <- seq(-6, 6, length.out = 200)

df_link <- data.frame( x = x, Logit = 1 / (1 + exp(-x)), Probit = pnorm(x) )

ggplot(df_link, aes(x = x)) + geom_line(aes(y = Logit, color = "Logit")) + geom_line(aes(y =
```

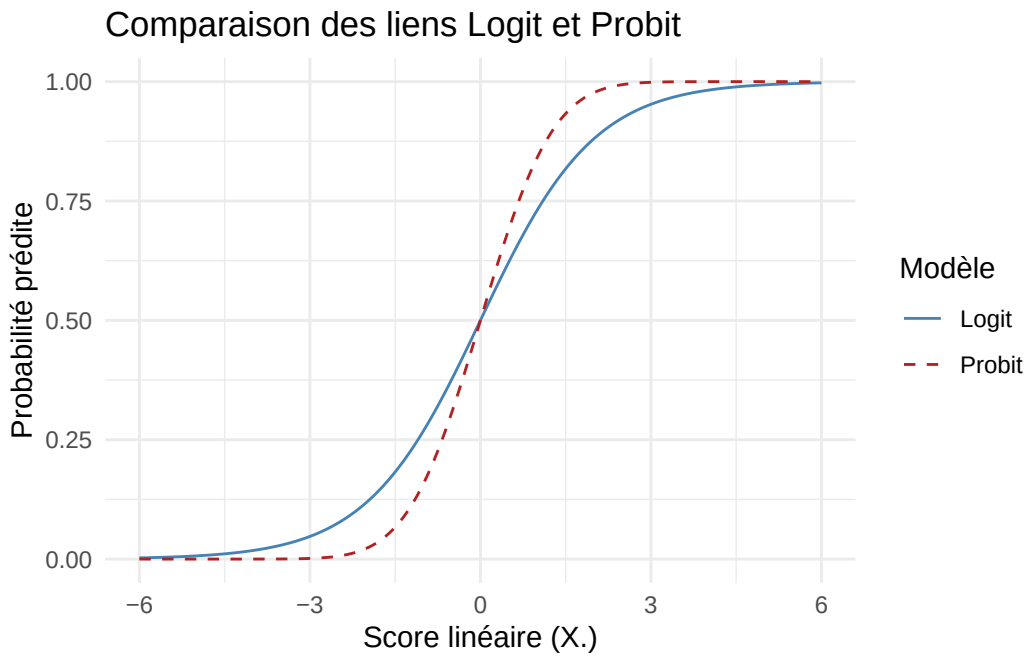


Figure 1: Comparaison des fonctions de lien Logit et Probit

Observation :

- Les deux courbes sont très proches ; la principale différence réside dans la forme légèrement plus aplatie du Probit aux extrémités.
- En pratique, les résultats logit et probit sont très similaires (seuls les coefficients changent d'échelle : $\beta_{\text{logit}} \approx 1.6\beta_{\text{probit}}$.

Interprétation économique

- **Logit** : privilégié quand on interprète les coefficients en termes d'odds-ratios (très courant en santé et sciences sociales).
- **Probit** : privilégié en économie microéconométrique, car il se relie naturellement à un modèle de variable latente et au Tobit.

```r
x <- seq(0, 40, length.out = 200)
beta0 <- -2.5; beta1 <- 0.12
p_hat <- 1 / (1 + exp(-(beta0 + beta1 * x)))

df_ex <- data.frame(debt_income = x, p_hat = p_hat)

ggplot(df_ex, aes(debt_income, p_hat)) +
geom_line(color = "seagreen4", linewidth = 1.2) +
labs(
title = "Effet du ratio dette/revenu sur la probabilité de défaut (modèle logit)",
x = "Ratio dette / revenu (%)",
y = "Probabilité prédite de défaut"
) +
theme_minimal()
```

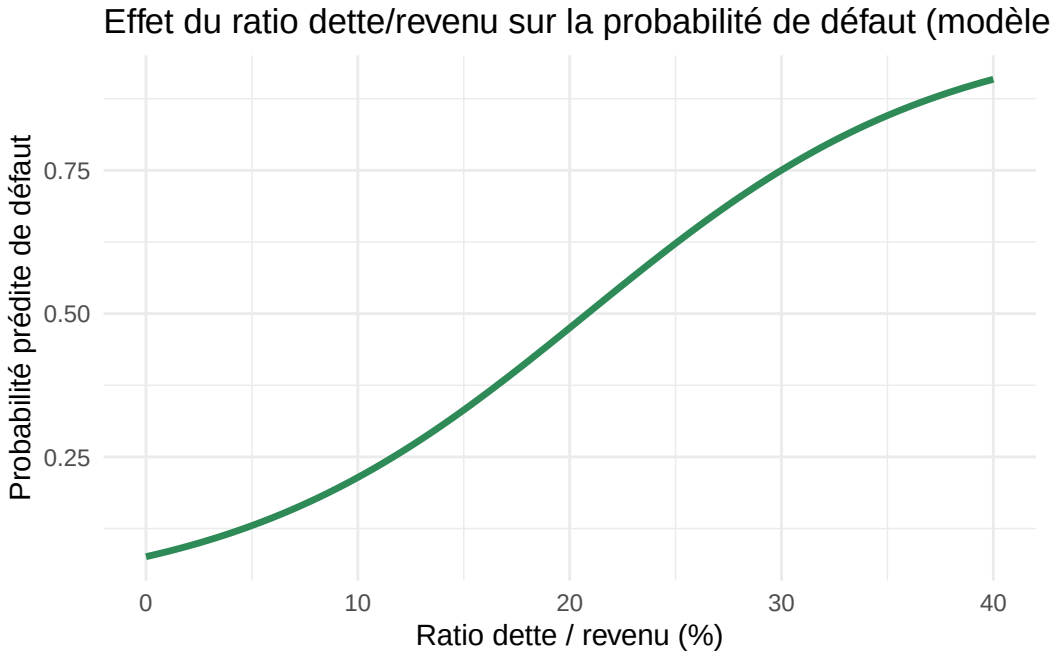


Figure 2: Exemple : probabilité prédite de défaut selon le ratio dette/revenu

Interprétation :

→ Lorsque le ratio dette/revenu augmente, la probabilité prédite de défaut croît de manière sigmoïde : faible au départ, elle augmente rapidement autour de la zone moyenne, puis se stabilise.

Résumé comparatif

Caractéristique	Logit	Probit
Fonction de lien	$\text{logit}(p) = \log(p/(1-p))$	$\Phi^{-1}(p)$
Distribution implicite des erreurs	Logistique	Normale centrée réduite
Interprétation des coefficients	Odds-ratios	Z-scores latents
Usages typiques	Santé, sociologie	Économie, finance
Résultats empiriques	Quasi identiques (_logit 1.6 _probit)	

À retenir : Logit et Probit sont deux manières voisines de modéliser une probabilité binaire. Le choix entre les deux est souvent une question de convention disciplinaire ou d'interprétabilité des coefficients.

Questions TD

Télécharger le fichier bankloanT.xls depuis l'ENT, puis importer les données dans R

```
df0 <- read_excel("./data/bankloanT.xls") |> clean_names()
df0 |> glimpse()
```

```
Rows: 850
Columns: 9
$ age      <dbl> 44, 26, 47, 31, 33, 45, 45, 35, 38, 32, 36, 47, 34, 39, 27, 4~
$ ed       <chr> "College degree", "High school degree", "Some college", "Did ~
$ employ   <dbl> 18, 6, 16, 5, 10, 21, 16, 17, 7, 0, 4, 23, 16, 8, 7, 8, 9, 0,~
$ address  <dbl> 23, 6, 7, 7, 2, 26, 21, 4, 4, 4, 17, 11, 9, 0, 8, 18, 6, 5, 1~
$ income   <dbl> 78, 30, 266, 23, 54, 132, 80, 42, 64, 20, 25, 115, 79, 21, 30~
$ debtinc  <chr> "1", "1", "2", "2", "3", "3", "3", "3", "3", "3", "4", "4", "~
$ creddebt <chr> "0.56472", "0.1437", "2.19184", "0.046", "0.11988", "2.55816"~
$ othdebt  <chr> "0.21528", "0.1563", "3.12816", "0.414", "1.50012", "1.40184"~
$ default  <chr> NA, "No", NA, NA, NA, "No", "No", "No", "No", "Yes", NA, "No"~
```

Etudier la distribution de la variable ed. Créer une variable catégorielle, puis une variable ne contenant que 4 classes.

```
df0 |> count(ed) |> arrange(desc(n))
```

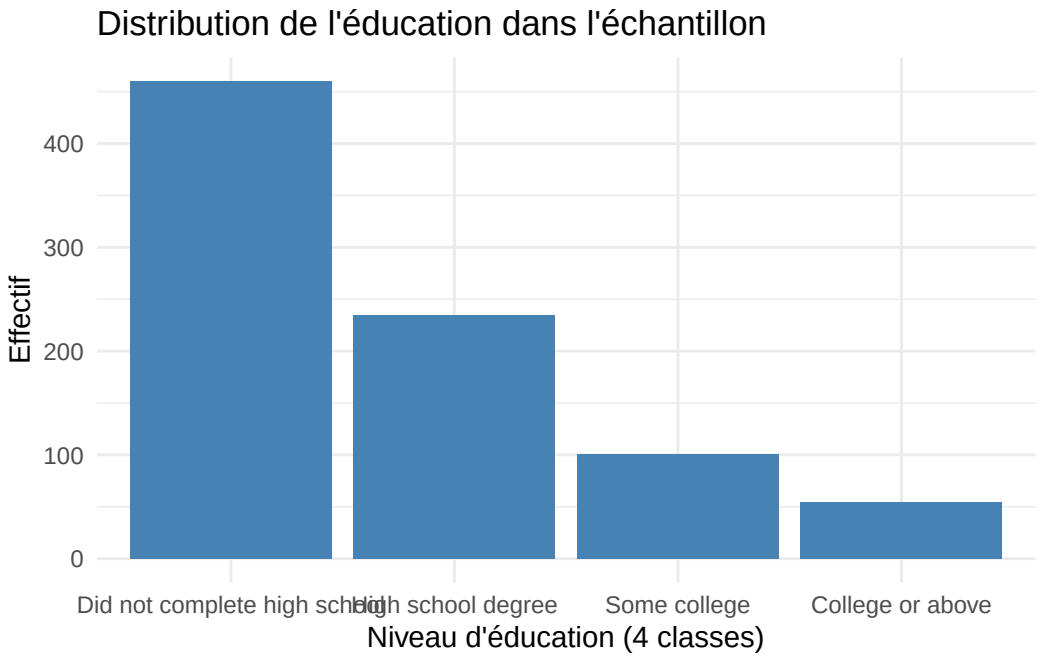
```
# A tibble: 5 x 2
  ed                              n
  <chr>                       <int>
1 Did not complete high school   460
2 High school degree             235
3 Some college                   101
```

```
4 College degree                   49
5 Post-undergraduate degree         5
```

```r
df1 <- df0 |>
  mutate(
    ed5 = factor(ed,
                 levels = c(
                   "Did not complete high school",
                   "High school degree",
                   "Some college",
                   "College degree",
                  "Post-undergraduate degree"
                 ),
                 ordered = TRUE),
    ed4 = fct_collapse(ed5,
                       "College or above" = c("College degree", "Post-undergraduate degree"))
  )

df1 |> count(ed4)
```

```
# A tibble: 4 x 2
  ed4                              n
  <ord>                        <int>
1 Did not complete high school   460
2 High school degree             235
3 Some college                   101
4 College or above                54
```

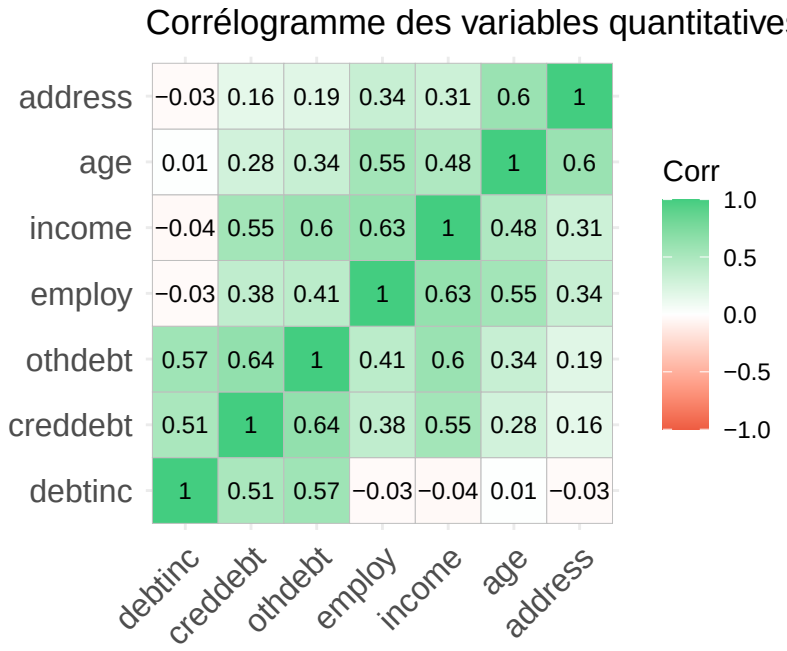
```r
ggplot(df1, aes(x = ed4)) +
  geom_bar(fill = "steelblue") +
  labs(x = "Niveau d'éducation (4 classes)", y = "Effectif",
       title = "Distribution de l'éducation dans l'échantillon") +
  theme_minimal()
```



Etudier la matrice des corrélations entre les 7 variables quantitatives présentes.

```
num_vars <- c("age","employ","address","income","debtinc","creddebt","othdebt")
df1 <- df1 |>
  mutate(
    debtinc  = as.numeric(debtinc),
    creddebt = as.numeric(creddebt),
    othdebt  = as.numeric(othdebt)
  )
cor_mat <- df1 |>
  select(all_of(num_vars)) |>
  drop_na() |>
  cor()

library(ggcorrplot)
ggcorrplot(
  cor_mat,
  hc.order = TRUE,           # ordonne les variables par similarité
  lab = TRUE,                # affiche les coefficients
  lab_size = 3,
  colors = c("tomato2", "white", "seagreen3"),
  title = "Corrélogramme des variables quantitatives",
```

```
  ggtheme = theme_minimal()
)
```



Recoder la variable à expliquer (default) en variable quantitative puis réaliser une première régression logistique incluant toutes les variables explicatives quantitatives.

```
df2 <- df1 |>
  mutate(
    default_num = case_when(
      default == "Yes" ~ 1,
      default == "No"  ~ 0,
      TRUE ~ NA_real_   # conserve les NA existants
    )
  )
form1 <- default_num ~ age + employ + address + income + debtinc + creddebt + othdebt
mod1 <- glm(form1, data=df2, family=binomial("logit"))
summary(mod1)
```

```
Call:
```

```
glm(formula = form1, family = binomial("logit"), data = df2)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.377693   0.571585  -2.410   0.0159 *
age          0.033694   0.017341   1.943   0.0520 .
employ      -0.265035   0.031996  -8.283  < 2e-16 ***
address     -0.103964   0.023193  -4.483 7.37e-06 ***
income      -0.007530   0.008099  -0.930   0.3525
debtinc      0.065253   0.030620   2.131   0.0331 *
creddebt     0.628263   0.113738   5.524 3.32e-08 ***
othdebt      0.070289   0.077693   0.905   0.3656
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 804.36  on 699  degrees of freedom
Residual deviance: 552.21  on 692  degrees of freedom
  (150 observations deleted due to missingness)
AIC: 568.21

Number of Fisher Scoring iterations: 6
```

Réaliser une deuxième régression logistique incluant aussi la variable educ en classe. Enregistrer aussi les résultats de ce modèle, qui est le modèle le plus complexe que nous appliquerons à ces données.

```r
form2 <- update(form1, . ~ . + ed4)
mod2 <- glm(form2, data=df2, family=binomial("logit"))
summary(mod2)
```

```
Call:
glm(formula = form2, family = binomial("logit"), data = df2)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.469161   0.585721  -2.508   0.0121 *
age          0.036527   0.017564   2.080   0.0376 *
```

```
employ       -0.259784   0.033353  -7.789 6.76e-15 ***
address      -0.105959   0.023331  -4.542 5.58e-06 ***
income       -0.007386   0.007927  -0.932   0.3515
debtinc       0.071049   0.030620   2.320   0.0203 *
creddebt      0.616294   0.112296   5.488 4.06e-08 ***
othdebt       0.052860   0.078374   0.674   0.5000
ed4.L         0.003376   0.321415   0.011   0.9916
ed4.Q        -0.334133   0.276144  -1.210   0.2263
ed4.C        -0.030154   0.249875  -0.121   0.9039
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 804.36  on 699  degrees of freedom
Residual deviance: 550.03  on 689  degrees of freedom
  (150 observations deleted due to missingness)
AIC: 572.03

Number of Fisher Scoring iterations: 6
```

Tester l'ajustement de ce modèle complet, grâce à la commande hoslem.test, où g est le nombre de groupes de niveaux différents de fonction prédictive utilisé pour le test. Varier les valeurs de g pour vérifier la robustesse du résultat.

```
hoslem.test(mod2$y, fitted(mod2), g=4)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  mod2$y, fitted(mod2)
X-squared = 2.3796, df = 2, p-value = 0.3043
```

```
hoslem.test(mod2$y, fitted(mod2), g=5)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  mod2$y, fitted(mod2)
X-squared = 0.9883, df = 3, p-value = 0.8041
```

```
hoslem.test(mod2$y, fitted(mod2), g=6)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  mod2$y, fitted(mod2)
X-squared = 5.3862, df = 4, p-value = 0.2499
```

```
hoslem.test(mod2$y, fitted(mod2), g=7)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  mod2$y, fitted(mod2)
X-squared = 2.0796, df = 5, p-value = 0.838
```

```
hoslem.test(mod2$y, fitted(mod2), g=8)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  mod2$y, fitted(mod2)
X-squared = 3.9178, df = 6, p-value = 0.6878
```

```
hoslem.test(mod2$y, fitted(mod2), g=9)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  mod2$y, fitted(mod2)
X-squared = 5.5538, df = 7, p-value = 0.5927
```

```
hoslem.test(mod2$y, fitted(mod2), g=10)
```

```
    Hosmer and Lemeshow goodness of fit (GOF) test

data:  mod2$y, fitted(mod2)
X-squared = 5.9712, df = 8, p-value = 0.6505
```

Grâce à la commande anova, réaliser un test de rapport de vraisemblance entre les deux modèles ajustés, et conclure sur la significativité de la variable educ.

```
anova(mod1, mod2, test="Chisq")
```

```
Analysis of Deviance Table

Model 1: default_num ~ age + employ + address + income + debtinc + creddebt +
    othdebt
Model 2: default_num ~ age + employ + address + income + debtinc + creddebt +
    othdebt + ed4
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       692     552.21
2       689     550.03  3    2.176   0.5367
```

- La p-value = **0.54** → on **ne rejette pas** H_0.
- Autrement dit :

  > **l'ajout de la variable d'éducation `ed4` n'améliore pas significativement la qualité du modèle.**

Ôter la variable la moins significative du modèle retenu. Vérifier que la P-value du test

de rapport de vraisemblance entre les modèles avec et sans cette variable est égale ou très proche de la P-value du test bilatéral de nullité du coefficient associé à la variable.

```
form3 <- update(form1, . ~ . - othdebt)
mod3 <- glm(form3, data=df2, family=binomial("logit"))
summary(mod3)
```

```
Call:
glm(formula = form3, family = binomial("logit"), data = df2)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.591125   0.522271  -3.047  0.00231 **
age          0.033618   0.017383   1.934  0.05312 .
```

```
employ       -0.257986   0.030791  -8.379  < 2e-16 ***
address      -0.103119   0.023141  -4.456 8.34e-06 ***
income       -0.002526   0.006320  -0.400  0.68939
debtinc       0.086173   0.020071   4.293 1.76e-05 ***
creddebt      0.595490   0.104930   5.675 1.39e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 804.36  on 699  degrees of freedom
Residual deviance: 553.02  on 693  degrees of freedom
  (150 observations deleted due to missingness)
AIC: 567.02

Number of Fisher Scoring iterations: 6
```

```r
anova(mod2, mod3, test="Chisq")
```

```
Analysis of Deviance Table

Model 1: default_num ~ age + employ + address + income + debtinc + creddebt +
    othdebt + ed4
Model 2: default_num ~ age + employ + address + income + debtinc + creddebt
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       689     550.03
2       693     553.02 -4  -2.9856   0.5602
```

Une variable est à nouveau très peu significative. Ajuster un nouveau modèle sans cette variable. Sauvegarder les valeurs prédites par ce modèle.

```r
form4 <- update(form3, . ~ . - income)
mod4 <- glm(form4, data=df2, family=binomial("logit"))
summary(mod4)
```

```
Call:
glm(formula = form4, family = binomial("logit"), data = df2)
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.63128    0.51268  -3.182  0.00146 **
age          0.03256    0.01717   1.896  0.05799 .
employ      -0.26076    0.03011  -8.662  < 2e-16 ***
address     -0.10365    0.02309  -4.490 7.13e-06 ***
debtinc      0.08926    0.01855   4.813 1.49e-06 ***
creddebt     0.57265    0.08723   6.565 5.20e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 804.36  on 699  degrees of freedom
Residual deviance: 553.18  on 694  degrees of freedom
  (150 observations deleted due to missingness)
AIC: 565.18

Number of Fisher Scoring iterations: 6
```

```r
anova(mod3, mod4, test="Chisq")
```

```
Analysis of Deviance Table

Model 1: default_num ~ age + employ + address + income + debtinc + creddebt
Model 2: default_num ~ age + employ + address + debtinc + creddebt
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       693     553.02
2       694     553.18 -1 -0.15877   0.6903
```

Etablir le tableau de contingence des individus bien ou mal classés avec une règle de coupure à 0,5

```r
# Probabilités prédites
df2_nona <- df2 |> drop_na(age, employ, address, income, debtinc, creddebt, othdebt, default

p_hat <- predict(mod4, newdata = df2_nona, type = "response")

df2_nona <- df2_nona |>
  mutate(
```

```
    p_hat = p_hat,
    pred_class = if_else(p_hat >= 0.5, 1, 0)
  )


table(Predicted = df2_nona$pred_class, Observed = df2_nona$default_num)
```

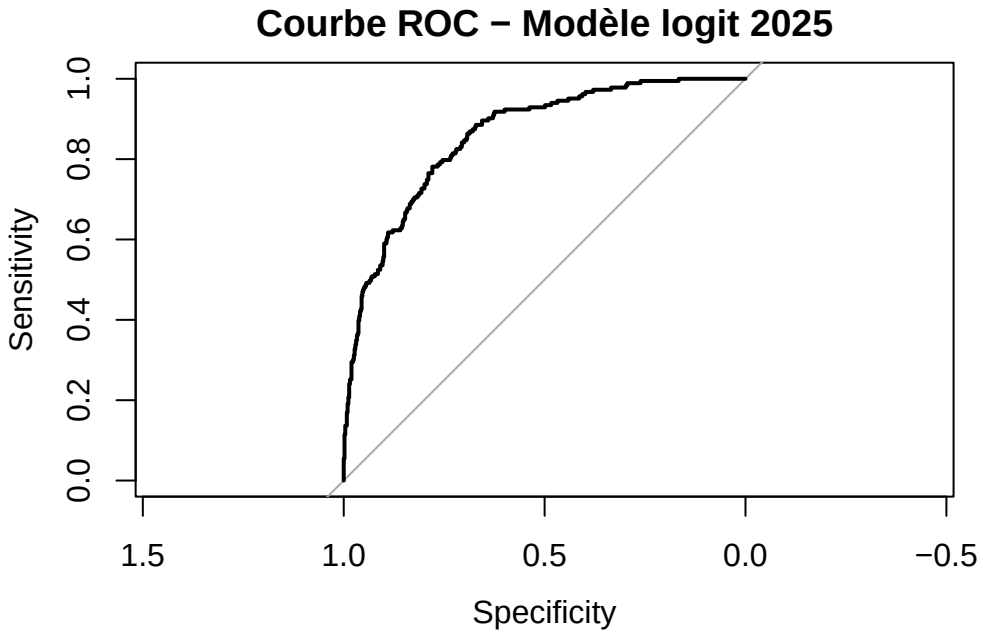
```
         Observed
Predicted   0   1
        0 476  89
        1  41  94
```

```
mean(df2_nona$pred_class == df2_nona$default_num, na.rm = TRUE)
```

```
[1] 0.8142857
```

Etablir la courbe ROC pour le « meilleur » modèle, puis calculer les "probabilités prédites" et étudier leur distribution

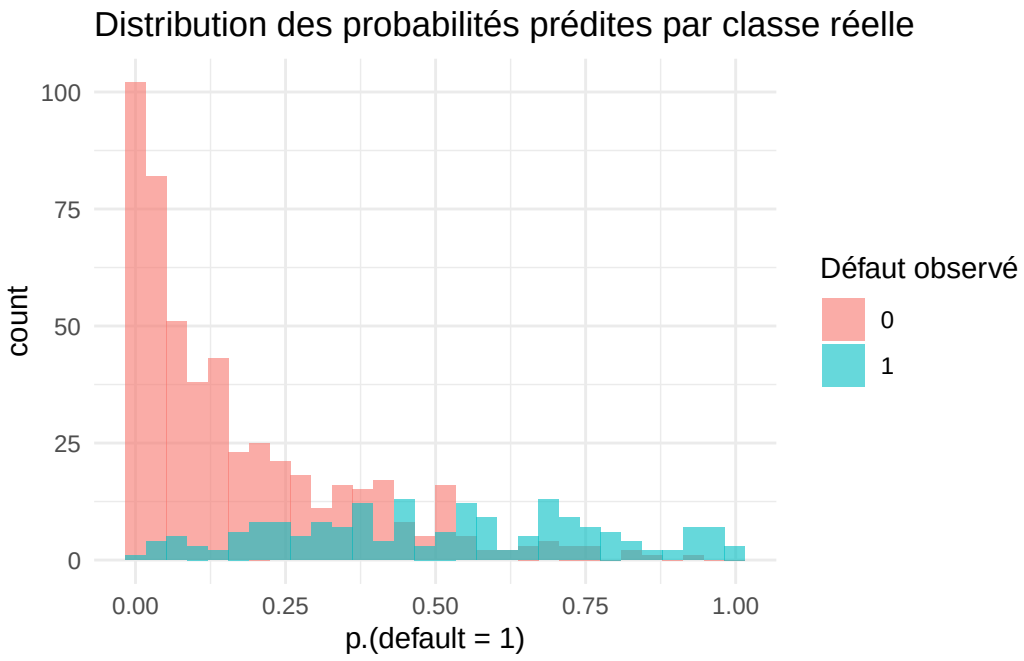
```
roc_obj <- roc(df2_nona$default_num, df2_nona$p_hat)

plot(roc_obj, main="Courbe ROC - Modèle logit 2025")
```



```r
auc(roc_obj)
```

```
Area under the curve: 0.8582
```

```r
ggplot(df2_nona, aes(x = p_hat, fill = factor(default_num))) +
  geom_histogram(alpha = 0.6, position = "identity", bins = 30) +
  labs(title = "Distribution des probabilités prédites par classe réelle",
       x = "p̂(default = 1)", fill = "Défaut observé") +
  theme_minimal()
```



Interprétation du graphe

- L'axe des abscisses montre les **probabilités prédites de défaut** $\hat{p} = P(\text{default} = 1|X)$.
- L'axe des ordonnées montre le **nombre d'individus** dans chaque intervalle de probabilité.
- La couleur rouge (0) représente les **emprunteurs qui n'ont pas fait défaut**.
- La couleur bleue (1) représente les **emprunteurs qui ont fait défaut**.

Refaire la même modélisation avec un modèle Probit et observer les différences et ressemblances avec la modélisation Logit.

```
mod_probit <- glm(formula(mod4), data=df2, family=binomial("probit"))
summary(mod_probit)
```

```
Call:
glm(formula = formula(mod4), family = binomial("probit"), data = df2)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.963869   0.297087  -3.244  0.00118 **
age          0.018237   0.009972   1.829  0.06742 .
employ      -0.144955   0.016217  -8.939  < 2e-16 ***
address     -0.055609   0.012835  -4.333 1.47e-05 ***
debtinc      0.051049   0.010563   4.833 1.35e-06 ***
creddebt     0.322172   0.048056   6.704 2.03e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 804.36  on 699  degrees of freedom
Residual deviance: 554.57  on 694  degrees of freedom
  (150 observations deleted due to missingness)
AIC: 566.57

Number of Fisher Scoring iterations: 6
```

Pour aller plus loin...

Supposant qu'un individu ne remboursant pas son emprunt coûte en moyenne 100000$, et qu'un individu payant son emprunt rapporte en moyenne 40000$, on peut calculer (...) qu'il est optimal de n'accorder un prêt qu'aux individus ayant une probabilité de rembourser estimée à 0,7 ou plus.

Règle de décision (probabilité de remboursement >= 0.7)

```
df2_nona <- df2_nona |> mutate(p_repay = 1 - p_hat, grant = as.numeric(p_repay >= 0.7))
mean(df2_nona$grant, na.rm = TRUE)
```

```
[1] 0.64
```

Etablir le tableau de contingence des individus bien ou mal classés en considérant comme défaillants potentiels tous les individus ayant moins de 70% de chances de rembourser. Commenter ce tableau.

```
thr_default <- 0.3
table(Predicted = df2_nona$p_hat >= thr_default, Observed = df2_nona$default_num)
```

```
         Observed
Predicted   0   1
    FALSE 405  43
    TRUE  112 140
```

```
tab <- table(Predicted = df2_nona$p_hat >= thr_default,
             Observed = df2_nona$default_num)
accuracy <- sum(diag(tab)) / sum(tab)
accuracy
```

```
[1] 0.7785714
```

```
prop.table(tab, 2)   # pourcentage par classe observée
```

```
         Observed
Predicted         0         1
    FALSE 0.7833656 0.2349727
    TRUE  0.2166344 0.7650273
```

En fixant le seuil à 0,3 (c'est-à-dire en considérant comme « défaillant potentiel » tout individu ayant moins de 70 % de chances de rembourser), le modèle devient **plus prudent** : il classe davantage d'individus comme risqués. Le nombre de vrais

positifs (défaillants correctement détectés) augmente, mais au prix d'une hausse des faux positifs (bons payeurs injustement rejetés).

Autrement dit, la règle minimise les pertes dues aux impayés, mais réduit le volume de prêts accordés. C'est un **compromis classique entre risque de crédit et rentabilité** :

Plus le seuil est bas, plus la banque protège son portefeuille, mais plus elle refuse de bons clients.