INFORME D4

0. Taula de continguts

1. Introducció

2. Dades del problema (D2)

3. Pla de treball (D3)

4. Objectius i metodologia

5. Preprocessament de les dades

Aquesta part correspon a l'script `preprocessing.rmd`. Als tres primers punts de l'script netegem noms de columnes, i recodifiquem múltiples variables a factors amb nivells i etiquetes coherents, després comprovem valors buits i duplicitats i generem un informe exploratori inicial amb `SmartEDA`, amb el qual ens guiarem a l'hora d'actuar als següents passos. També verifiquem files amb NA i files duplicades.

Al punt 4, eliminem *outliers*, basant-nos en el que hem pogut veure a l'informe general. En concret, es marquen:

- **Edat en el moment de la inscripció (`Age_at_enrollment`):** superior a 30 anys.
- **Ordre de sol·licitud (`Application_order`):** superior a 7.
- **Nota de la qualificació prèvia (`Previous_qualification_grade`):** superior a 170.
- **Nota d'admissió (`Admission_grade`):** superior a 150.
- **Nombre d'unitats curriculars (1r i 2n semestre):**
  - Matriculades (`*_enrolled`): superior a 10.
  - Avaluades (`*_evaluations`): superior a 10.
  - Aprovades (`*_approved`): superior a 10.

Aquest procés garanteix la coherència i la plausibilitat de les dades, eliminant valors inusuals o impossibles que podrien distorsionar les anàlisis posteriors. Per exemple, per la variable `Admission_grade`, hem pogut observar els següents canvis:

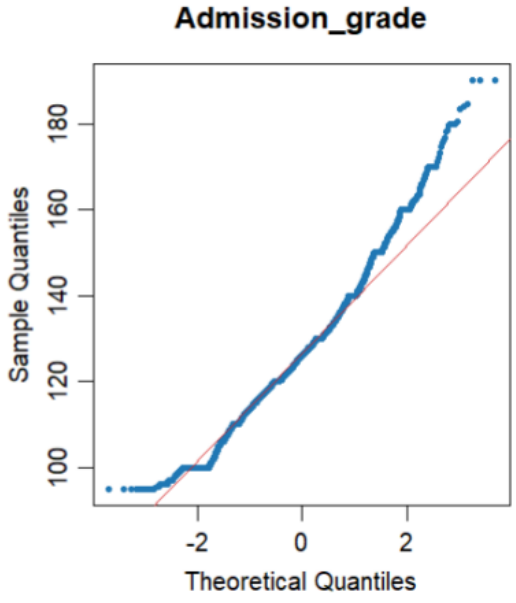

Figura 1: Variable numèrica admisson_grade abans del preprocessament

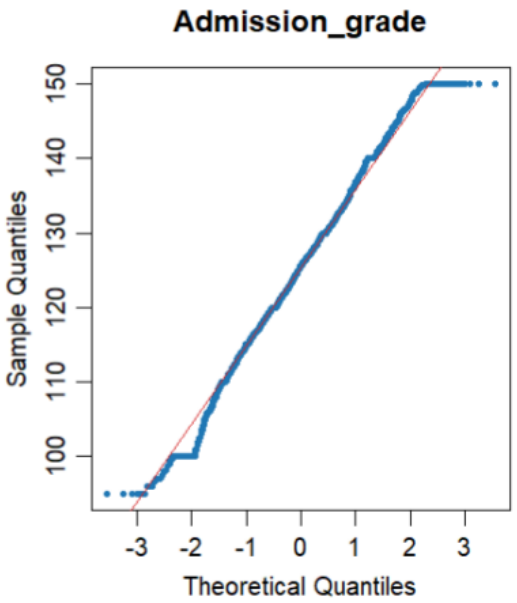

Figura 2: Variable numèrica admisson_grade després del preprocessament

En el punt 5 i amb ajuda del nostre informe inicial, hem identificat diverses variables que representen quantitats discretes. Aquestes variables s'han transformat en variables categòriques amb valors compresos entre 0 i 10. Les variables afectades són les següents:

- `Application_order`

- `Curricular_units_1st_sem_enrolled`
- `Curricular_units_2nd_sem_enrolled`
- `Curricular_units_1st_sem_evaluations`
- `Curricular_units_2nd_sem_evaluations`
- `Curricular_units_1st_sem_approved`
- `Curricular_units_2nd_sem_approved`

També, hem processat les variables relacionades amb els crèdits i les avaluacions pendents per convertir-les en variables binàries (on qualsevol valor superior a 0 es codifica com "1"):

- `Curricular_units_1st_sem_credited`
- `Curricular_units_2nd_sem_credited`
- `Curricular_units_1st_sem_without_evaluations`
- `Curricular_units_2nd_sem_without_evaluations`

Aquests canvis permeten una millor interpretació i anàlisi de les dades, facilitant l'aplicació de tècniques estadístiques i models predictius que requereixen variables categòriques. Finalment, al punt 6, hem guardat el dataset netejat i preprocessat en un nou fitxer CSV anomenat `clean-data.csv`, que serà utilitzat per a les anàlisis posteriors.

6. Anàlisi exploratori

cal canviar Quan mirem el conjunt de dades, veiem que hi ha diferents tipus de variables, algunes numèriques i d'altres categòriques. En general, les numèriques no segueixen del tot una forma "normal", algunes tenen valors molt agrupats i d'altres tenen punts que surten bastant del que seria esperable. Els gràfics de densitat i els boxplots mostren que hi ha bastants valors atípics i que les distribucions són força diferents entre variables, cosa que indica que no totes representen la informació de la mateixa manera. En alguns casos, hi ha variables amb una dispersió molt alta i altres molt concentrades, cosa que pot complicar una mica les anàlisis o la creació de models. Pel que fa a les variables categòriques, podem veure que no estan gaire equilibrades. Hi ha algunes categories molt freqüents, mentre que en tenim d'altres que gairebé no apareixen. Això pot fer que els resultats no siguin del tot justos, perquè el model podria donar més importància a les categories que tenen més dades i passar per alt les que en tenen poques, creant un biaix que s'hauria de tenir en compte a l'hora de fer models o prediccions.

Quan mirem les relacions entre variables, podem veure que n'hi ha algunes que estan bastant relacionades. Els boxplots mostren diferències clares entre grups i els scatterplots deixen veure certs patrons que podrien ser útils més endavant. Tot això, ens fa pensar que hi ha variables amb una influència real sobre la variable resposta. En resum, el conjunt de dades és interessant, amb tendències clares i relacions útils, però també amb desequilibris i valors extrems que caldrà tenir en compte abans de fer models més avançats o prediccions.

7. Ajustament d'un MLGz numèric

El nostre objectiu és modelar i validar un model lineal generalitzat, amb `Admission_grade` com a resposta numèrica. Partim carregant les nostres dades netejades i preprocessades, assegurant-nos que totes les variables estan en el format correcte. Comencem fent un histograma de la variable resposta per entendre la seva distribució i identificar possibles transformacions necessàries amb la comanda `hist(data$Admission_grade)`:

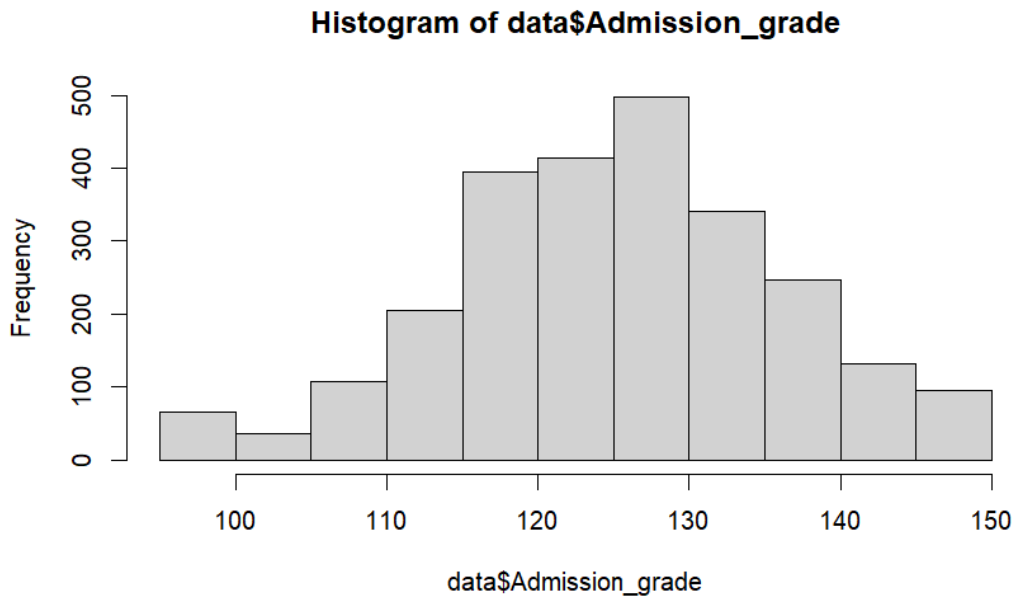

Figura 3: Histograma de la variable Admisssion_grade

Continuem fent un ajustament inicial d'un model lineal generalitzat amb la família gaussiana i el link identitat, utilitzant totes les variables excepte la variable 'Target'. Aquest model ens servirà com a punt de partida per a la selecció de variables i l'avaluació del rendiment. El construïm el model amb la comanda següent:

```
model <- glm(Admission_grade ~ . - Target, data = data, family = gaussian())
```

També construïm un model nul per tenir una referència bàsica i amb l'objectiu de comparar-lo amb el model complet. El model nul només inclou l'intercept i ens permet avaluar la millora que aporta el model amb totes les variables. El construïm amb la comanda següent:

```
model_nul <- glm(Admission_grade ~ 1, data = data, family = gaussian)
```

Executem un test òmnibus o ANOVA per comparar el model complet amb el model nul, utilitzant `anova(model_nul, model, test="Chisq")`. Aquest test ens ajudarà a determinar si el model complet ofereix una millora significativa en la predicció de la variable resposta en comparació amb el model nul. Obtenim els resultats següents:

```
  Resid. Df Resid. Dev   Df Deviance  Pr(>Chi)
1      2389     145767
2      2534     307211 -145  -161444 < 2.2e-16 ***
```

Veiem que el nostre model redueix la deviància en 161444 respecte el model nul, i el test de Chi-quadrat dona un p-valor extremadament petit. Això vol dir que el nostre model amb variables explicatives millora significativament l'ajust.

7.1 Selecció de variables

Per a començar a buscar el nostre model final, caldrà que eliminem les variables que no aporten informació rellevant i volem eliminar variables redundants, per quedar-nos amb un model més simple que s'ajusti de forma similar. Per això, ens guiarem amb els resultats que ens dona la pròpia construcció del model complet, on podem veure en consola les variables més explicatives i les que tenen menys pes, que són:

- Variables amb significació molt alta (***): `Application_mode`, `Course`, `Previous_qualification`, `Previous_qualification_grade`, `Age_at_enrollment`, `Unemployment_rate`.
- Variables amb significació alta (**): `Application_order`, `Curricular_units_1st_sem_evaluations`, `GDP`.

Aquest model l'anomenarem `model_1`, que, comparant-lo amb el model complet amb un altre test ANOVA, ens dóna els següents resultats:

```
  Resid. Df Resid. Dev  Df Deviance  Pr(>Chi)
1      2489     155505
2      2389     145767 100   9738.5 0.0001416 ***
```

Podem plantejar les següents hipòtesis:

- **Hipòtesi nul·la H_0:** El model simple (`model_1`) és suficient. Les variables addicionals del model complet no milloren l'ajust de manera significativa.
- **Hipòtesi alternativa H_1:** El model complet (`model_complet`) ajusta les dades significativament millor que el model simple.

Els resultats mostren que l'augment de la deviància és de 9738.5 (que ja en si ens fa veure que les variables excloses en el model simplificat tenen, en conjunt, un poder explicatiu significatiu) amb un p-valor de 0.0001416, que és molt petit. Això ens porta a rebutjar l'hipòtesi nul·la i acceptar l'hipòtesi alternativa. Per tant, el model complet ajusta les dades significativament millor que el model simple. Descartem doncs, el model simple i seguim amb el model complet.

Però necessitem un model més senzill, així que seguim eliminant variables amb poca significació. El trobarem amb la funció `step()` de R, que ens ajuda a fer una selecció automàtica de variables basada en el criteri d'informació d'Akaike (AIC). Aquesta funció prova diferents combinacions de variables, afegint o eliminant-les, i selecciona el model que minimitza l'AIC, que és una mesura que equilibra la qualitat de l'ajust amb la complexitat del model. Així, podem trobar un model que sigui tant precís com senzill. Llavors, executem la comanda següent: `model_final <- step(model, direction = "both")`

Aquest procés ens porta a un model final que anomenarem `model_final`, que inclou només les variables que aporten informació rellevant per predir `Admission_grade`. Aquest model és més manejable i interpretable, mantenint al mateix temps una bona capacitat predictiva. Un cop obtenim aquest model final, el comparem amb el model complet utilitzant un altre test ANOVA per veure si la simplificació ha afectat significativament l'ajust del model. Els resultats són els següents:

```
  Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
1      2389     145767
2      2465     150821 -76  -5054.3   0.2768
```

També tornem a plantejar les següents hipòtesis:

- **Hipòtesi nul·la H_0:** El model final (`model_final`) és suficient. Les variables addicionals del model complet no milloren l'ajust de manera significativa.
- **Hipòtesi alternativa H_1:** El model complet (`model_complet`) ajusta les dades significativament millor que el model final.

Els resultats mostren que la reducció de la deviància és de -5054.3 amb un p-valor de 0.2768, que és molt gran. Això ens porta a no rebutjar l'hipòtesi nul·la. Per tant, el model final ajusta les dades de manera similar al model complet. Així doncs, acceptem el model final com a model definitiu per predir `Admission_grade`. I per acabar-nos-en d'assegurar, mirem la puntuació AIC dels dos models:

- **AIC del model complet:** 17759.37
- **AIC del model final:** 17693.77

Veiem que l'AIC del model final és menor que el del complet, la qual cosa reforça la nostra decisió d'escollir el model final. Un AIC més baix indica un millor equilibri entre l'ajust del model i la seva complexitat, fent que el model final sigui preferible per a la predicció de `Admission_grade`.

7.2 Validació del model

Per a la validació del nostre model, visualitzem diferents gràfics i els analitzem:

- **Gràfic de residus vs. valors ajustats:** Aquest gràfic ens ajuda a veure si hi ha algun patró en els residus. Si els residus estan distribuïts aleatòriament al voltant de zero, això indica que el model s'ajusta bé. Si veiem algun patró, podria indicar que el model no està capturant alguna relació important. Executem la comanda `residualPlot(model_final)`, de la llibreria `car` per generar aquest gràfic:

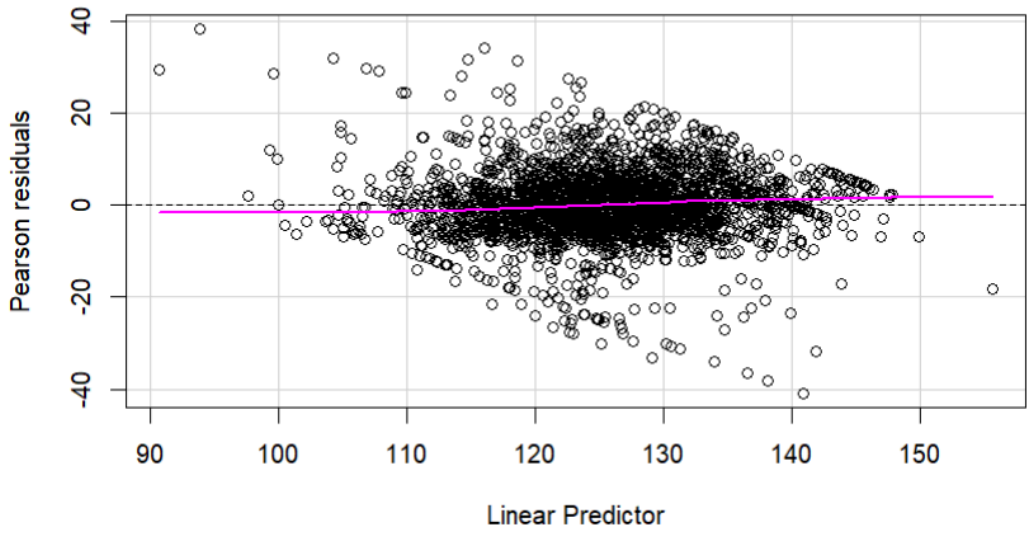

Figura 4: Gràfic de residus vs. valors ajustats

El gràfic de residus de Pearson contra el predictor lineal suggereix un ajust global raonable: la dispersió és força simètrica al voltant de 0 i la corba suau resta gairebé horitzontal, cosa coherent amb variància aproximadament constant en l'escala del link i absència de patrons forts visibles. Es detecten algunes observacions extremes i una lleugera inflexió als extrems de l'eix x, que podrien indicar petites no linealitats localitzades o efectes de límit, però no s'hi aprecia un patró en embut clar d'heteroscedasticitat. Amb aquestes observacions doncs, podem dir que el model sembla ajustar raonablement bé la major part de les dades.

- **Altres gràfics de residus:** Generem també gràfics addicional de residus contra cadascun dels predictors de primer ordre del model, utilitzant residus de tipus Pearson, per comprovar si hi ha algun patró específic relacionat amb alguna variable en particular. Utilitzem la comanda `residualPlots(model_final)` de la llibreria `car` per generar aquests gràfics:

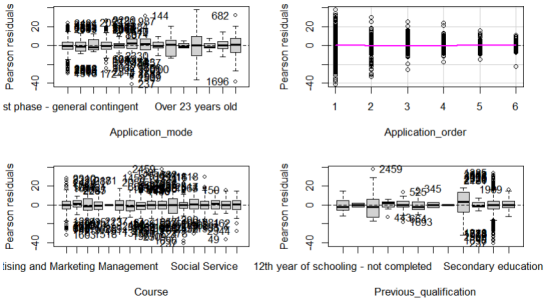

Figura 5: Residus de Pearson varis

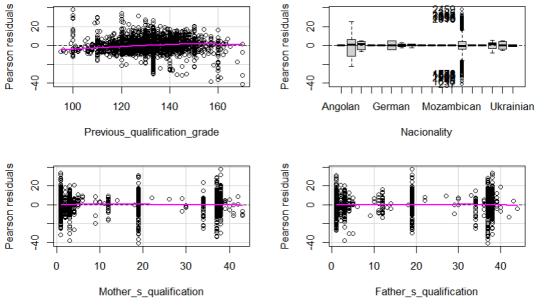

Figura 6: Residus de Pearson varis

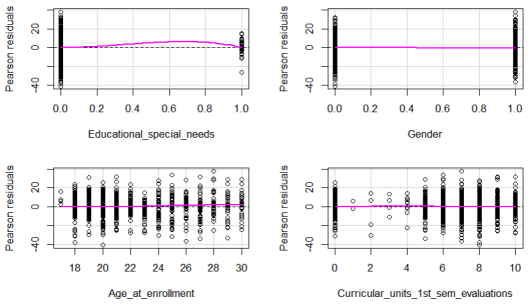

Figura 7: Residus de Pearson varis

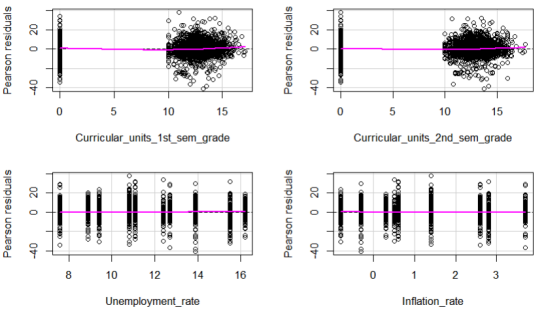

Figura 8: Residus de Pearson varis

Pel que fa aquests gràfics, el patró global és compatible amb un ajust raonable del GLM: la línia suau es manté molt a prop de 0 en la majoria de panells i els residus es dispersen de manera bastant simètrica, sense evidència clara d'heteroscedasticitat en forma d'embut ni de curvatures marcades persistents. Podem concloure que:

- Per les **variables contínues** de rendiment acadèmic: als panells de `Curricular_units_1st_sem_grade`, `Curricular_units_2nd_sem_grade` i `Previous_qualification_grade`, la línia suau resta propera a 0 i la nuvolositat és densa però sense tendències fortes; això suggereix que el link capta raonablement la relació mitjana amb aquestes covariables, malgrat alguns punts extrems que convé revisar com a potencials outliers.
- Pels **indicadors macro** com `Unemployment_rate`, `Inflation_rate`: es veuen bandes verticals (per valors repetits) amb variància relativament estable i línia suau plana; no s'aprecien patrons sistemàtics de biaix en aquests predictors i, per tant, no hi ha senyals clares d'heteroscedasticitat associada al nivell d'aquests índexs.
- Pels **factors categòrics grans** `Application_mode`, `Course`, `Previous_qualification`, `Nacionality`: la línia de referència es manté al voltant de 0 a la majoria de nivells; hi ha alguns grups amb pocs casos o dispersió desigual que mostren caixes més amples o valors atípics, però en general no es detecten patrons sistemàtics de desviació que indiquin problemes d'ajust.

En resum, els gràfics no revelen violacions greus; el model sembla ben especificat en l'escala del link, amb possibles millores menors centrades en la gestió d'outliers.

- **InfluencePlot de la llibreria `car`**: La llibreria `car` també ofereix la funció `influencePlot()` per visualitzar l'impacte de les observacions individuals en el model. Aquest gràfic ens dona 3 mesures clau: l'eix x són els hat values, és a dir, l'apalancament de cada observació; l'eix y són els residus estandarditzats; i la mida dels punts representa la distància de Cook, que mesura l'efecte global d'eliminar una observació en els paràmetres del model. Executem la comanda `influencePlot(model_final)` per generar aquest gràfic:

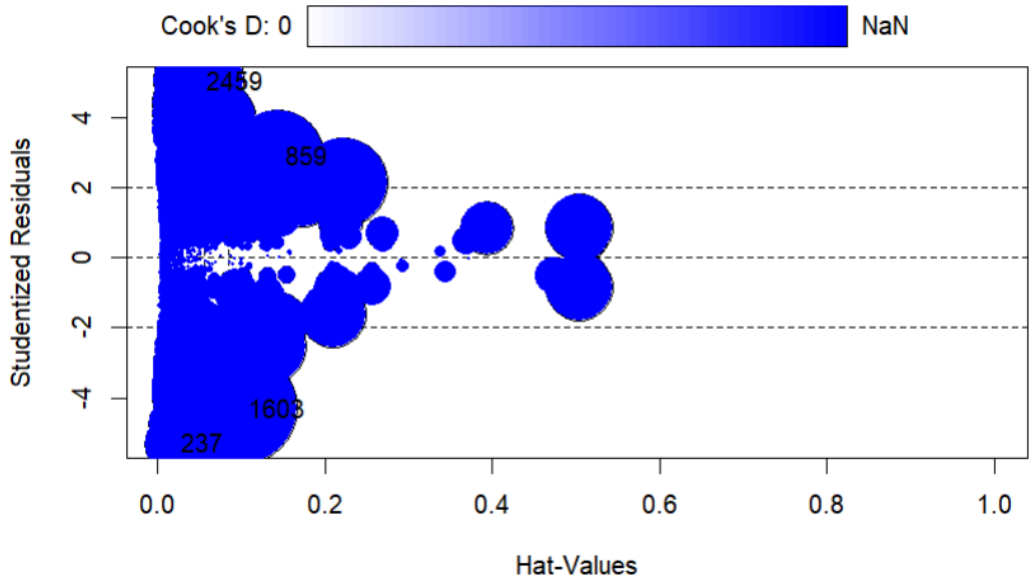

Figura 9: Influence Plot del model final

La majoria d'observacions tenen leverage molt baix (hat-value pròxim a 0) i residus dins de l'interval $[-2, 2]$, per tant no són problemàtiques ni per error ni per palanca, és a dir, la combinació de predictors d'aquella observació és "central" dins el núvol de dades Un grup petit de punts apareix cap a hat-values entre 0.2 i 0.6 amb residus moderats; aquestes bombolles són més grans, indicant distància de Cook més alta i, per tant, potencial influència sobre els coeficients del model. Els punts etiquetats: 2459, 859, 1608, 237, han superat algun llindar "noteworthy" en residu, leverage o Cook's D i, per això, el car els marca automàticament; aquests són candidats prioritaris a revisió del registre, valors extrems o combinacions inusuals de predictors. Tot així, cap d'aquests punts no té un leverage extremadament alt (per exemple, > 0.8) ni residus fora de l'interval $[-4, 4]$, per tant no semblen ser influències desproporcionades que distorsionin greument l'ajust global del model.

Per tant, com a conclusió de la validació, el model sembla ajustar bé les dades sense violacions greus dels supòsits. Hi ha algunes observacions amb certa influència que convé revisar, però en general el model és robust i adequat per a la predicció de `Admission_grade`.

8. Ajustament d'un MLGz binari

L'objectiu d'aquest apartat és ajustar un model lineal generalitzat (MLGz) per a una variable resposta binària, on la resposta representa si l'estudiant abandona (dropout) o segueix/gradua (enrolled o graduated). Com que la variable `Target` té tres categories (enrolled, graduated, dropout), la convertirem en una variable binària per a aquest model específic. Assignarem el valor 1 a `dropout` i el valor 0 a les altres dues categories (`enrolled` i `graduated`):

```
base$target <- ifelse(x == "dropout", 1L,
                              ifelse(x %in% c("enrolled","graduated"), 0L, 0L))
```

Per començar, fem un anàlisi exploratori de la nova variable resposta binària `target` en funció de les possibles variables explicatives. Utilitzem gràfics de barres per a les variables categòriques i boxplots per a les variables numèriques, per veure com es distribueixen els valors de `target` en funció d'aquestes variables. Això ens ajudarà a identificar quines variables podrien tenir una influència significativa en la probabilitat d'abandonament. En color verd tenim els estudiants que no abandonen (target = 0) i en vermell els que abandonen (target = 1).

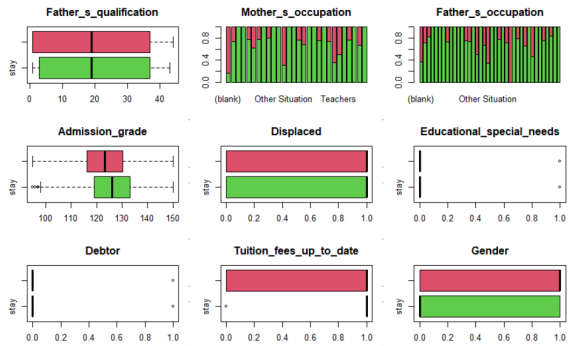

Figura 10: Anàlisi exploratori

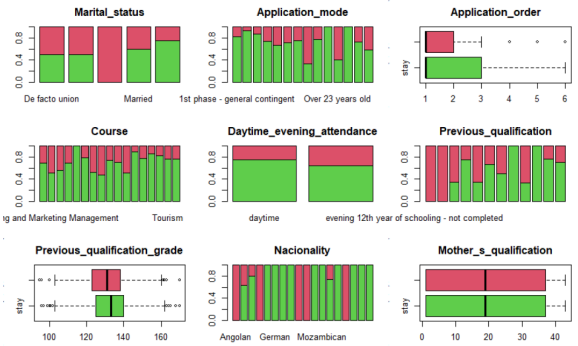

Figura 11: Anàlisi exploratori

En destaquem dos fets:

- Sembla que hi ha variables que semblarien ser més rellevants per predir l'abandonament, com ara `Previous_qualification_grade` o `Admission_grade`, on es veu que els estudiants que abandonen tenen notes més baixes en aquestes variables. Això podria indicar que el rendiment acadèmic és un factor important en la decisió d'abandonar els estudis.
- Trobem que algunes categories de variables com `Course`, `Mother_s_occupation` o `Father_s_occupation`, entre d'altres, tenen proporcions molt altres o del 100%, cosa que podria provocar separació completa en el model i dificultar l'ajustament. Haurem de tenir-ho en compte a l'hora de seleccionar les variables per al model i hauríem de considerar eliminar o agrupar aquestes categories per evitar problemes d'ajustament.

Un cop vist això, procedim a ajustar un model lineal generalitzat amb la família binomial i el link logit, utilitzant totes les variables explicatives disponibles, és a dir, ajustant el model complet. Aquest model ens permetrà estimar la probabilitat d'abandonament en funció de les diferents característiques dels estudiants. També construïm un model nul per tenir una referència bàsica i amb l'objectiu de comparar-lo amb el model complet. El model nul només inclou l'intercept i ens permet avaluar la millora que aporta el model amb totes les variables.

El construïm amb la comanda següent:

```
model_complet <- glm(target ~ ., data = base, family = binomial(link = "logit"))
model_nul <- glm(target ~ 1, data = base, family = binomial(link = "logit"))
```

Fent això però, obtenim un missatge d'advertència que ens indica:

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Aquest missatge ens indica que, a l'hora d'ajustar el model, s'han trobat probabilitats ajustades que són exactament 0 o 1. És a dir, per algunes combinacions de les variables explicatives, el model prediu amb total certesa que un estudiant abandonarà o no abandonarà els estudis. Com sospitavem a l'anàlisi exploratori, això pot ser degut a la presència de categories amb proporcions molt extremes o del 100% en algunes variables, cosa que provoca separació completa en el model. Aquesta situació pot dificultar l'ajustament del model i afectar la seva capacitat predictiva. Intentem diagnosticar doncs, si hi ha separació o quasi-separació en les dades i on, i si en trobem, intentem solucionar-ho d'alguna manera

8.1 Diagnòstic i solució de separació

Per a veure si les nostres dades tenen separació o quasi-separació, utlitzarem la funció `detect_separation()` de la llibreria `detectseparation`. Aquesta funció ens permet identificar si hi ha problemes de separació en les dades que poden afectar l'ajustament del model i en quines variables es produeixen aquests problemes. Aquesta funció requereix de la matriu de covariables de totes les variables

explicatives i del vector de la variable resposta binària, és a dir, els valors de la variable target. Executem la comanda següent:

```
X <- model.matrix(target ~ ., data = base) # Matriu de covariables
y <- base$target
det <- detect_separation(X, y, family = binomial())
```

L'objecte `det` conté informació sobre la presència de separació en les dades. Podem examinar els resultats per veure si hi ha problemes de separació i quines variables estan implicades. Segons aquest objecte, trobem que hi ha separació completa en les següents variables:

- `Application_mode`
- `Course`
- `Previous_qualification`
- `Nacionality`
- `Mother_s_occupation`
- `Father_s_occupation`
- `Marital_status`

Com podem veure, són variables amb un alt nombre de categories. Per tant, pensem que la separació es deu a la presència de categories amb poques observacions i que totes tenen el mateix valor de la variable resposta. Una possible solució per evitar aquest problema és agrupar les categories amb poques observacions en una categoria anomenada "Other". D'aquesta manera, podem reduir el nombre de categories i evitar que hi hagi categories amb poques observacions que puguin causar separació completa en el model. Implementem aquesta possible solució fent ús de la funció `fct_lump_min()` de la llibreria `forcats`, que ens permet agrupar les categories amb menys d'un nombre mínim d'observacions en una categoria anomenada "Other". Agrupem amb un llindar de 150 observacions per a cada variable problemàtica, per tal d'assegurar-nos que cada categoria tingui una representació suficient en les dades. Apliquem aquesta funció a cadascuna de les variables problemàtiques:

```
Application_mode = fct_lump_min(Application_mode, min = 150, other_level = "Other")
```

Fets aquests canvis, tornem a provar de diagnosticar la separació en les dades utilitzant de nou la funció `detect_separation()`. Si ja no hi ha problemes de separació. Ara ens retorna que no hi ha separació completa en les dades, cosa que indica que la nostra solució ha estat efectiva. Ara que aparentment hem solucionat el problema de separació, podem procedir a ajustar el model lineal generalitzat.

8.2 Ajustament del model inicial

Ajustem de nou el model complet i el model nul amb les dades modificades, utilitzant la mateixa comanda que abans:

```
model_complet <- glm(target ~ ., data = base, family = binomial(link = "logit"))
model_nul <- glm(target ~ 1, data = base, family = binomial(link = "logit"))
```

Executem un test ANOVA comparant el model complet amb el model nul per veure si el model complet ofereix una millora significativa en la predicció de la variable resposta en comparació amb el model nul. Per tant, executem la comanda `anova(model_nul, model_complet, test="Chisq")`. Obtenim els resultats següents:

```
  Resid. Df Resid. Dev  Df Deviance  Pr(>Chi)
1      2456     1333.4
2      2534     2884.1 -78  -1550.6 < 2.2e-16 ***
```

Podem veure que el nostre model redueix la deviància en 1550.6 respecte el model nul, i el test de Chi-quadrat dona un p-valor extremadament petit. Això vol dir que el nostre model amb variables explicatives millora significativament l'ajust.

Del model complet, executem un test òmnibus o ANOVA amb test de ràtio de versemblança amb la comanda `anova(model_complet, test = "LR")` per obtenir una visió general de la significació global del model. Dels resultats obtingut escollirem els predictors més rellevants per al nostre model final. Els predictors seleccionats són els següents:

- `Course`
- `Application_mode`
- `Mother_s_qualification`
- `Father_s_occupation`
- `Debtor`
- `Tuition_fees_up_to_date`
- `Scholarship_holder`
- `Age_at_enrollment`
- `Curricular_units_1st_sem_approved`
- `Curricular_units_2nd_sem_enrolled`
- `Curricular_units_2nd_sem_approved`
- `Unemployment_rate`

S'han seleccionat aquests predictors perquè, d'una banda, mostren una contribució clara a la reducció de la deviància segons proves de raó de versemblança (variables amb grans caigudes de deviance i p-valors molt petits com `Course`, `Application_mode`, `Debtor`, `Tuition_fees_up_to_date` i els indicadors de rendiment acadèmic `Curricular_units_*`, especialment els approved), i, de l'altra, permeten un model estable evitant redundàncies dins de blocs altament correlacionats; així, es prioritzen approved del 1r i 2n semestre com a mesures més informatives de progrés, es manté només un indicador de càrrega del 2n semestre (enrolled) per no solapar-se amb "evaluations", i s'inclouen factors administratius robustos (`Debtor`, `Tuition_fees_up_to_date`, `Scholarship_holder`) junt amb variables demogràfiques i de context amb efecte consistent (`Age_at_enrollment`, `Unemployment_rate`), mentre que predictors amb senyal feble o redundant es deixen fora excepte si milloren criteris d'informació (AIC/BIC) o són necessaris com a controls teòrics (p. ex., `Mother_s_qualification`, `Father_s_occupation`), verificant que la combinació final minimitzi la colinealitat i mantingui interpretabilitat per a contrastos concrets via comparacions jeràrquiques.

Doncs, el model inicial m0.1 serà:

```
m0.1 <- glm(target ~
                Course +
                Application_mode +
                Mother_s_qualification +
                Father_s_occupation +
                Debtor +
                Tuition_fees_up_to_date +
                Scholarship_holder +
                Age_at_enrollment +
                Curricular_units_1st_sem_approved +
                Curricular_units_2nd_sem_enrolled +
                Curricular_units_2nd_sem_approved +
                Unemployment_rate,
              data = base,
              family = binomial(link = "logit"))
```

Del que primer de tot fem un test ANOVA entre el model nul i aquest model inicial m0.1 per veure si hi ha una millora significativa:

```
  Resid. Df Resid. Dev  Df Deviance Pr(>Chi)
1      2456     1333.4
2      2487     1380.5 -31  -47.102  0.03202 *
```

Per a saber si aquest model és millor que el model complet, plantegem les següents hipòtesis:

- **Hipòtesi nul·la H_0:** El model inicial (`m0.1`) és suficient. Les variables addicionals del model complet no milloren l'ajust de manera significativa.
- **Hipòtesi alternativa H_1:** El model complet (`model_complet`) ajusta les dades significativament millor que el model inicial.

Com que els resultats mostren que la reducció de la deviància és de -47.102 amb un p-valor de 0.03202, que és menor que 0.05, hauríem de rebutjar l'hipòtesi nul·la i acceptar l'hipòtesi alternativa. Però com que no hi ha tanta diferència de deviància entre els dos models, i per tal de tenir un model més senzill, acceptem el model inicial `m0.1` com a model possible per a la predicció de l'abandonament.

Per assegurar-nos que aquest model és el millor, utilitzem la funció `step()` de R per fer una selecció automàtica de variables basada en el criteri BIC. Aquesta funció prova diferents combinacions de variables, afegint o eliminant-les, i selecciona el model que minimitza el BIC. Executem la comanda següent:

```
m0.2 <- step(model_complet, direction = "backward", trace = FALSE)
```

Un cop obtingut aquest model final `m0.2`, els comprarem amb el model inicial `m0.1` utilitzant les puntuacions AIC i BIC:

```
AIC(model_complet, m0.1, m0.2)
BIC(model_complet, m0.1, m0.2)
```

Dels que podem construir la següent taula de resultats:

Model	df	AIC	BIC
`model_complet`	54	1491.845	1807.094
`m0.1`	26	1478.341	1630.128
`m0.2`	34	1467.215	1665.705

El model `m0.2` té l'AIC més baix, amb una diferència de 11.126 respecte `m0.1` i de 24.630 respecte el complet, que es considera evidència clara a favor de `m0.2` en comparació de models basada en informació. Però en BIC, `m0.1` és lleugerament millor que `m0.2` per la penalització més severa a la complexitat. Com que l'objectiu és tenir un model predictiu precís però també manejable, i la diferència d'AIC és no és tant significativa, escollim `m0.1` com a model final per a la predicció de l'abandonament.

8.3 Validació del model

Començarem amb una validació gràfica del model: Generem gràfics de residus, tant per les variables categòriques com per les numèriques:

Amb la comanda `residualPlots(m0.1, terms = ~ 1, type = "pearson", fitted = TRUE)` obtenim gràfics de residus de Pearson contra els valors ajustats:

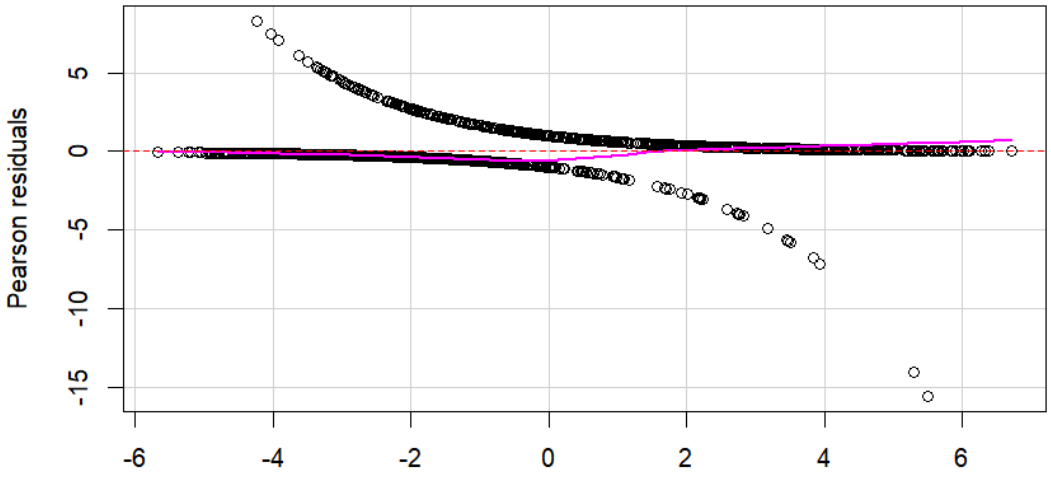

Figura 12: Gràfic de residus de Pearson vs. valors ajustats

En general,podem veure el patró central és força pla al voltant de 0 i no mostra tendències fortes, cosa coherent amb un ajust global raonable del GLM binomial amb link logit. Podem veure la presencia de dos valors amb residus alts, que poden indicar outliers o observacions amb certa influència, però en conjunt no hi ha patrons sistemàtics evidents que suggereixin problemes greus d'especificació del model. La dispersió dels residus sembla relativament estable al llarg de l'escala dels valors ajustats.

Pel que fa als residus per cada predictor, visualitzem els gràfics:

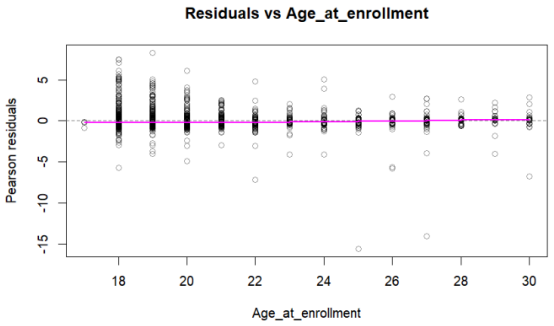

Figura 13: Residus de Pearson de la variable Age_at_enrollement

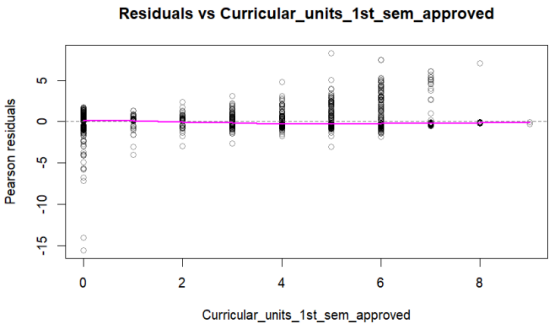

Figura 14: Residus de Pearson de la variable Curricular_units_1st_sem_approved

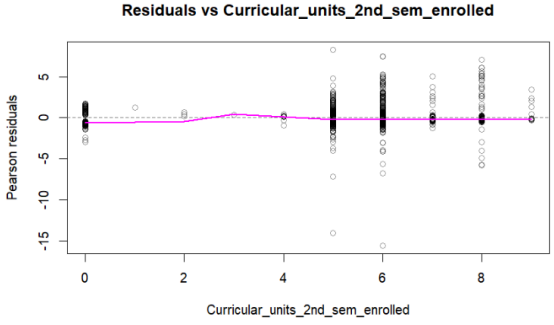

Figura 15: Residus de Pearson de la variable Curricular_units_2nd_sem_enrolled

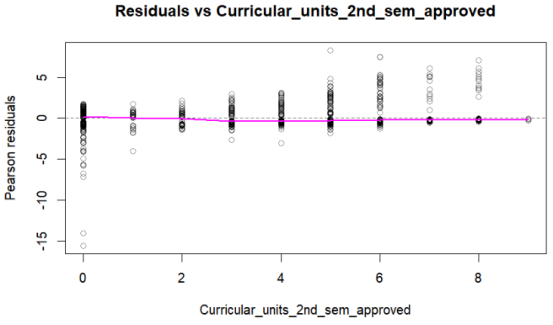

Figura 16: Residus de Pearson de la variable Curricular_units_2nd_sem_approved

No hi ha patrons sistemàtics, tot i que s'observa lleu no-linealitat en variables com Curricular_units_1st_sem_approved, Curricular_units_2nd_sem_enrolled i Curricular_units_2nd_sem_approved.

Mirem els boxplots de residus per a les variables categòriques:

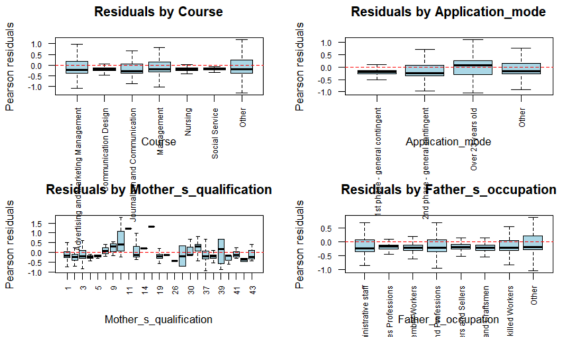

Figura 17: Residus de Pearson per variables categòriques

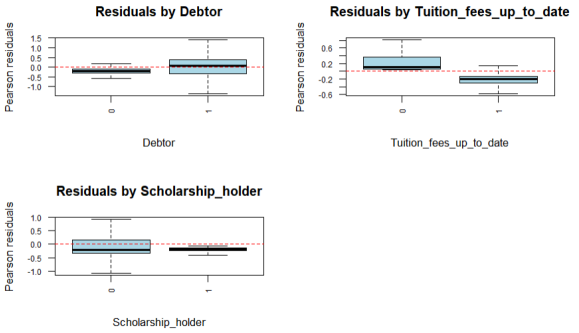

Figura 18: Residus de Pearson per variables categòriques

En general, els boxplots mostren que la majoria de les categories tenen residus centrats al voltant de 0, cosa que indica un bon ajust del model. No obstant això, algunes categories presenten una dispersió més gran o valors atípics, cosa que podria suggerir la necessitat d'una revisió addicional o d'una possible transformació de les variables.

Finalment, utilitzem la funció `influencePlot()` de la llibreria `car` per visualitzar l'impacte de les observacions individuals en el model:

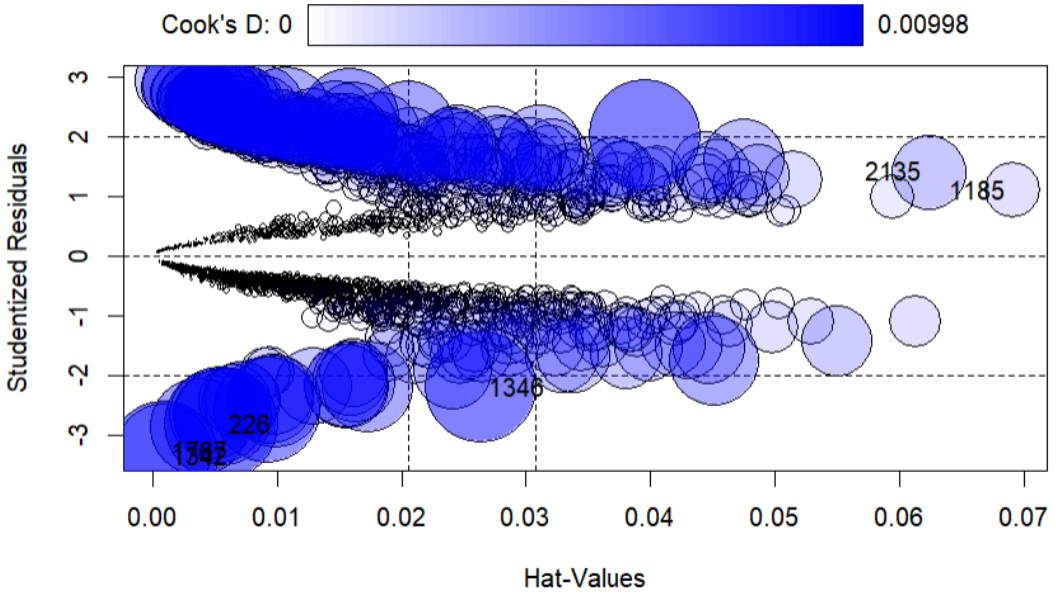

Figura 19: Influence Plot del model m0.1

La majoria d'observacions tenen leverage baix (< 0.03) i residus dins de l'interval aproximat $[-2, 2]$, amb valors de Cook's D_i petits (escala superior mostra màxim al voltant de 0.01), cosa que indica que no hi ha punts amb influència desmesurada sobre l'ajust global. Els punts etiquetats (p. ex., 226, 1346, 1185, 2135) destaquen per leverage o residus una mica més grans, però continuen per sota de llindars normals i, visualment, no creuen de manera clara línies de referència extremes; no semblen comprometre el model.

8.4 Possibles millores

Tot i que gràficament veiem que tenim un model prou bo, intentem millorar-lo de dues maneres

8.4.1 Dades agregades per binomis

Per les variables:

- `Curricular_units_1st_sem_approved`
- `Curricular_units_2nd_sem_enrolled`
- `Curricular_units_2nd_sem_approved`

Creem una base de dades agregada (df1, df2, df3), on s'han agrupat les observacions segons el valor de la variable corresponent. Això permet analitzar la relació entre la variable i la resposta binària (dropout sí/no) en una escala més agregada i estable estadísticament. Per a cada grup, calculem el nombre d'estudiants que han abandonat. L'objectiu és obtenir, per a cada valor possible de la variable, el nombre d'alumnes que han abandonat `ypos` i els que no han abandonat `yneg`:

```
  df1 <- with(base, aggregate(
  x = cbind(ypos = target, yneg = 1 - target),
  by = list(units1 = Curricular_units_1st_sem_approved),
  FUN = sum
))
```

Ajustem un model binomial amb aquesta nova base de dades agregada:

```
  m1 <- glm(cbind(ypos, yneg) ~ units1, data = df1, family = binomial(link='logit'))
```

Fem això per a les tres variables i analitzem els resultats. Mitjançant els gràfics de residus (`residualPlots(m1)`, `residualPlots(m2)`, `residualPlots(m3)`), comprovem visualment si aquesta relació és adequada o si apareixen patrons que suggereixen una forma corbada. En cas afirmatiu, podríem considerar afegir termes quadràtics o polinòmics per capturar millor la relació no lineal entre la variable i la probabilitat d'abandonament.

Per la variable `Curricular_units_1st_sem_approved`:

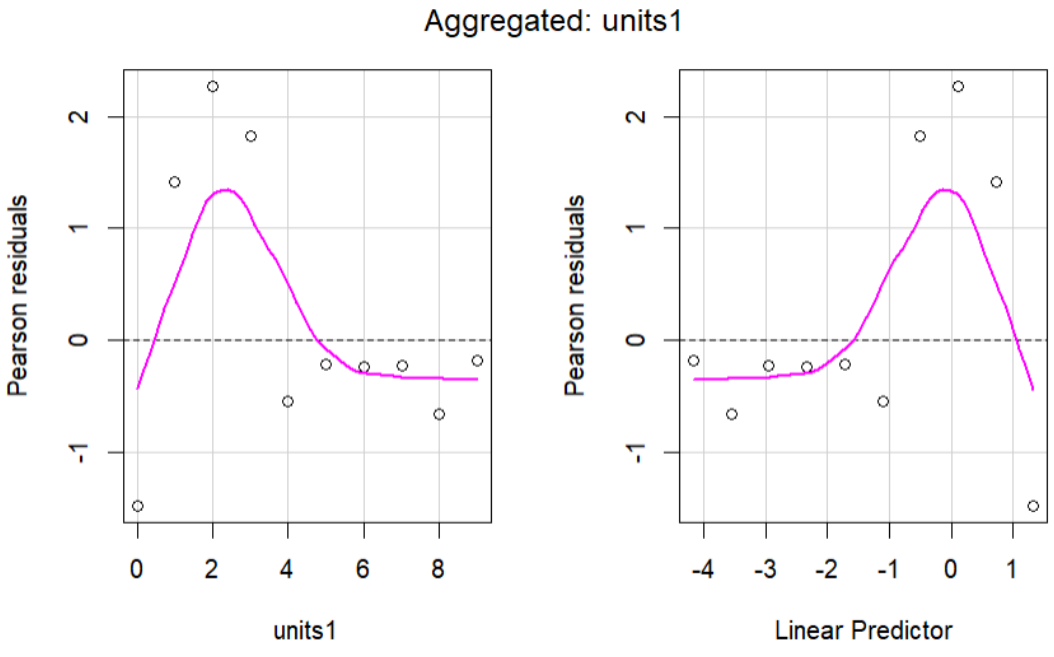

Figura 20: Gràfic de residus de Pearson vs. valors ajustats

Com que mostra un patró clarament no lineal, provem de descobrir quina pot ser la millor transformació. Provarem amb diferents termes polinòmics i veurem quin s'ajusta millor:

```
m1.1 <- glm(cbind(ypos,yneg) ~ units1,           data = df1, family = binomial)
m1.2 <- glm(cbind(ypos,yneg) ~ poly(units1, 2),  data = df1, family = binomial)
m1.3 <- glm(cbind(ypos,yneg) ~ poly(units1, 3),  data = df1, family = binomial)
m1.4 <- glm(cbind(ypos,yneg) ~ poly(units1, 4),  data = df1, family = binomial)
```

Comparant els AIC dels diferents models, trobem que el model amb terme polinòmic d'ordre 4 és el que té l'AIC més baix:

Model	df	AIC	BIC
m1.1	2	61.10337	61.70854
m1.2	3	58.97480	59.88255
m1.3	4	56.85660	58.06694
m1.4	5	54.11044	55.62336

També executem els tests Òmnibus consecutius per veure si cada model millora significativament respecte l'anterior:

```
anova(m1.1, m1.2, test = "Chisq")
anova(m1.2, m1.3, test = "Chisq")
anova(m1.3, m1.4, test = "Chisq")
```

Obtenim els següents resultats:

```
Model 1: cbind(ypos, yneg) ~ units1
Model 2: cbind(ypos, yneg) ~ poly(units1, 2)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         8    13.8119
2         7     9.6833  1   4.1286  0.04217 *
---------------------------------------------
Model 1: cbind(ypos, yneg) ~ poly(units1, 2)
Model 2: cbind(ypos, yneg) ~ poly(units1, 3)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         7     9.6833
2         6     5.5651  1   4.1182  0.04242 *
---------------------------------------------
Model 1: cbind(ypos, yneg) ~ poly(units1, 3)
Model 2: cbind(ypos, yneg) ~ poly(units1, 4)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         6     5.5651
2         5     0.8190  1   4.7462  0.02936 *
```

Observem que cada model millora significativament respecte l'anterior, ja que tots tenen p-valors menors que 0.05 i consecutivament menors. Per tant, si anem rebutjant hipòtesis nul·les H_0 que podem anar plantejant, escollim el model m1.4 amb terme polinòmic d'ordre 4 com a millor model per a la variable `Curricular_units_1st_sem_approved`.

Per la variable `Curricular_units_2nd_sem_enrolled`:

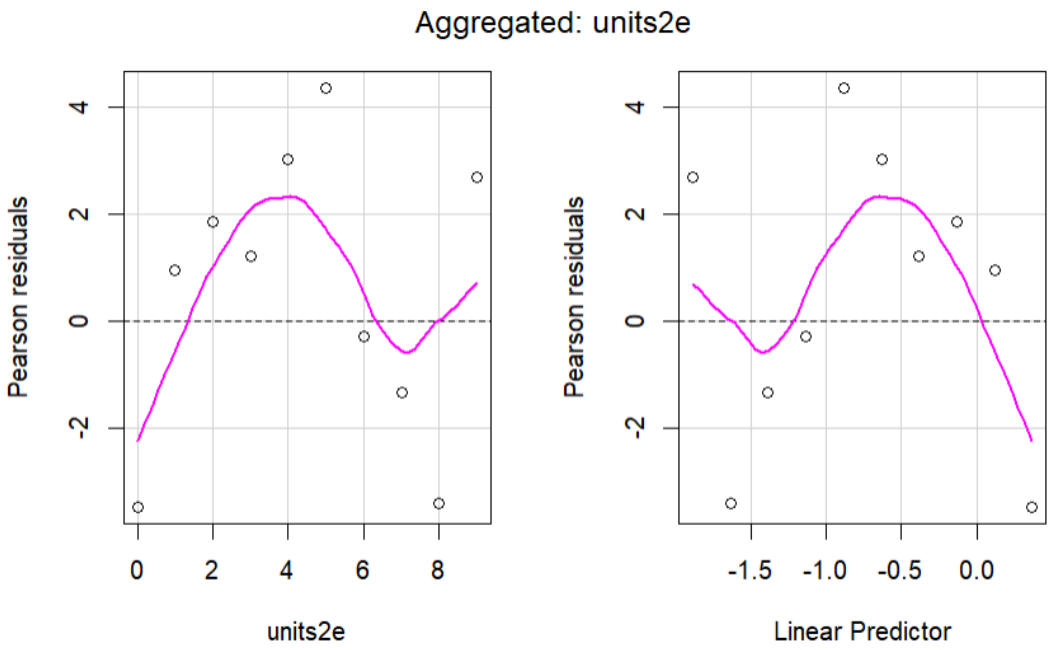

Figura 21: Gràfic de residus de Pearson vs. valors ajustats

Com que s'observa curvatura, provem termes polinòmics de diferent ordre i comparem quin s'ajusta millor:

```
m2.1 <- glm(cbind(ypos,yneg) ~ units2e, data = df2, family = binomial)
m2.2 <- glm(cbind(ypos,yneg) ~ poly(units2e, 2), data = df2, family = binomial)
m2.3 <- glm(cbind(ypos,yneg) ~ poly(units2e, 3), data = df2, family = binomial)
m2.4 <- glm(cbind(ypos,yneg) ~ poly(units2e, 4), data = df2, family = binomial)
```

Comparant AIC/BIC, el polinomi d'ordre 3 (m2.3) és el millor:

Model	df	AIC	BIC
m2.1	2	105.29389	105.89906
m2.2	3	72.62200	73.52975
m2.3	4	54.41665	55.62699
m2.4	5	56.38317	57.89610

També fem els tests Òmnibus entre models imbricats:

```
anova(m2.1, m2.2, test = "Chisq")
anova(m2.2, m2.3, test = "Chisq")
anova(m2.3, m2.4, test = "Chisq")
```

Obtenim els següents resultats:

```
Model 1: cbind(ypos, yneg) ~ units2e
Model 2: cbind(ypos, yneg) ~ poly(units2e, 2)
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1         8     66.099
2         7     31.427  1   34.672 3.902e-09 ***
---------------------------------------------
```

```
Model 1: cbind(ypos, yneg) ~ poly(units2e, 2)
Model 2: cbind(ypos, yneg) ~ poly(units2e, 3)
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1         7     31.427
2         6     11.222  1   20.205 6.956e-06 ***
---------------------------------------------
Model 1: cbind(ypos, yneg) ~ poly(units2e, 3)
Model 2: cbind(ypos, yneg) ~ poly(units2e, 4)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         6     11.222
2         5     11.188  1 0.033475   0.8548
```

Observem que els dos primers models milloren significativament respecte l'anterior, però el quart no aporta millora significativa respecte el tercer (p-valor alt). Per tant, escollim el model m2.3 amb terme polinòmic d'ordre 3 com a millor model per a la variable `Curricular_units_2nd_sem_enrolled`.

Per la variable `Curricular_units_2nd_sem_approved`:

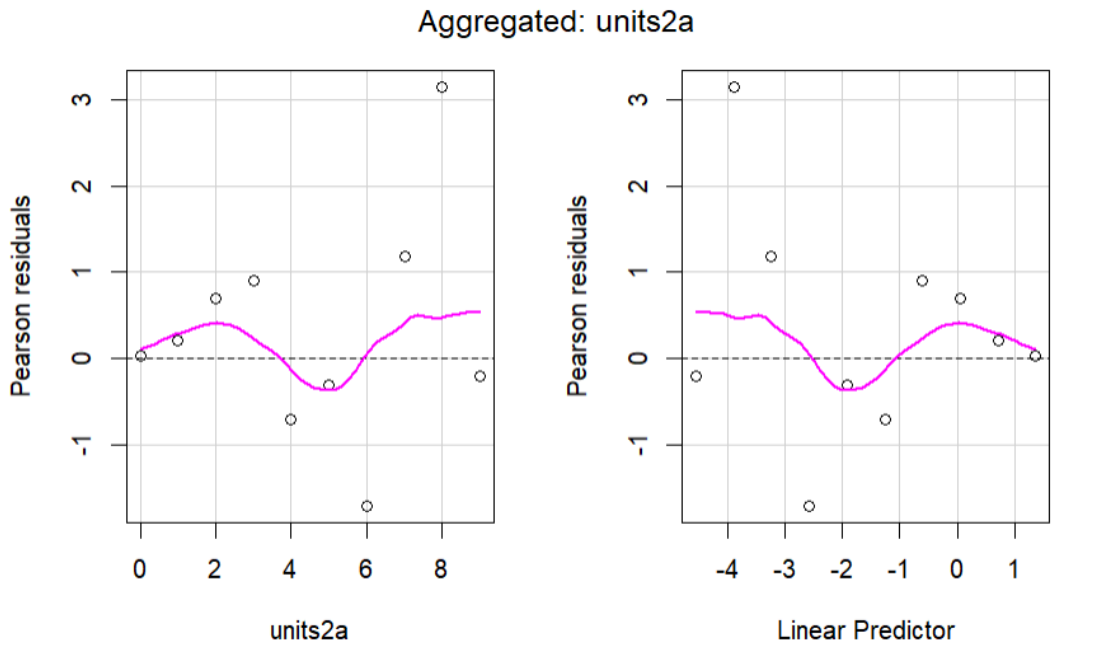

Figura 22: Gràfic de residus de Pearson vs. valors ajustats

S'exploren polinomis d'ordre creixent i es compara el seu ajust:

```
m3.1 <- glm(cbind(ypos,yneg) ~ units2a, data = df3, family = binomial)
m3.2 <- glm(cbind(ypos,yneg) ~ poly(units2a, 2), data = df3, family = binomial)
m3.3 <- glm(cbind(ypos,yneg) ~ poly(units2a, 3), data = df3, family = binomial)
m3.4 <- glm(cbind(ypos,yneg) ~ poly(units2a, 4), data = df3, family = binomial)
```

AIC/BIC indiquen que el cúbic (m3.3) és el millor:

Model	df	AIC	BIC
m3.1	2	62.23186	62.83703
m3.2	3	62.09034	62.99809
m3.3	4	55.77421	56.98455

m3.4	5	57.63726	59.15018

També es realitzen tests Òmnibus:

```
anova(m3.1, m3.2, test = "Chisq")
anova(m3.2, m3.3, test = "Chisq")
anova(m3.3, m3.4, test = "Chisq")
```

Obtenim els següents resultats:

```
  Model 1: cbind(ypos, yneg) ~ units2a
Model 2: cbind(ypos, yneg) ~ poly(units2a, 2)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
  1      8     13.547
  2      7     11.405  1  2.1415   0.1434
-----------------------------------------------
Model 1: cbind(ypos, yneg) ~ poly(units2a, 2)
Model 2: cbind(ypos, yneg) ~ poly(units2a, 3)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
  1      7     11.405
  2      6     3.089   1  8.3161   0.003929 **
-----------------------------------------------
Model 1: cbind(ypos, yneg) ~ poly(units2a, 3)
Model 2: cbind(ypos, yneg) ~ poly(units2a, 4)
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
  1     6      3.0894
  2     5      2.9524   1  0.13696 0.7113
```

Observem que només el pas del segon al tercer model és significatiu (p-valor < 0.05). Per tant, escollim el model m3.3 amb terme polinòmic d'ordre 3 com a millor model per a la variable `Curricular_units_2nd_sem_approved`.

Un cop tenim els tres models amb les transformacions adequades per a cada variable, tornem a visualitzar els gràfics de residus per assegurar-nos que ara no hi ha patrons sistemàtics.

Per a la variable `Curricular_units_1st_sem_approved`:

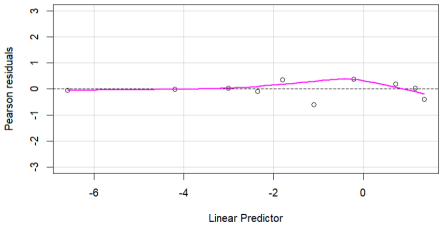

Figura 23: Residus ajustats per a la variable Curricular_units_1st_sem_approved

La línia magenta mostra una corba suau amb una sola pujada i baixada, i després s'estabilitza al voltant de zero. El quart ordre millora clarament el lineal; tot i no ser completament recta, la tendència principal s'ha corregit. És acceptable

Per a la variable `Curricular_units_2nd_sem_enrolled`:

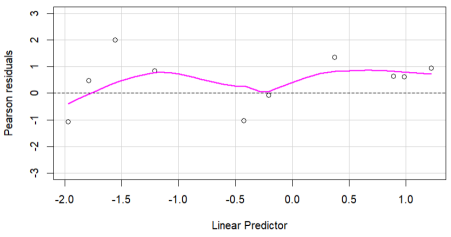

Figura 24: Residus ajustats per a la variable Curricular_units_2nd_sem_enrolled

Aquí la línia és més irregular, amb oscil·lacions o una doble corba. Indica que la relació és més complexa i el polinomi d'ordre 3 l'ha capturat massa ajustat, és a dir, n'ha fet el que es coneix com a overfitting. No és acceptable.

Per a la variable `Curricular_units_2nd_sem_approved`:

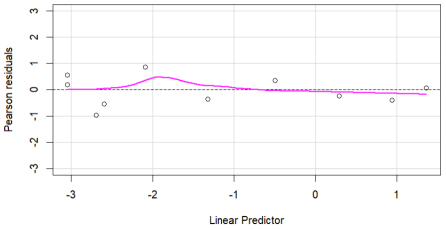

Figura 25: Residus ajustats per a la variable Curricular_units_2nd_sem_approved

Hi ha una forma lleu però consistent, sense observar patrons forts, amb una relació no estrictament lineal, però ben captada. Per tant, sembla adequat i acceptable

Per tant, composem un nou model final `m0.3` que inclogui només aquelles transformacions que considerem acceptables:

```
m0.3 <- glm(target ~
              Course +
              Application_mode +
              Mother_s_qualification +
              Father_s_occupation +
              Debtor +
              Tuition_fees_up_to_date +
              Scholarship_holder +
              Age_at_enrollment +
              poly(Curricular_units_1st_sem_approved, 4) +
              Curricular_units_2nd_sem_enrolled +
              poly(Curricular_units_2nd_sem_approved, 3) +
              Unemployment_rate,
            data = base,
            family = binomial(link = "logit"))
```

Generem els gràfics de residus per aquest nou model `m0.3`:

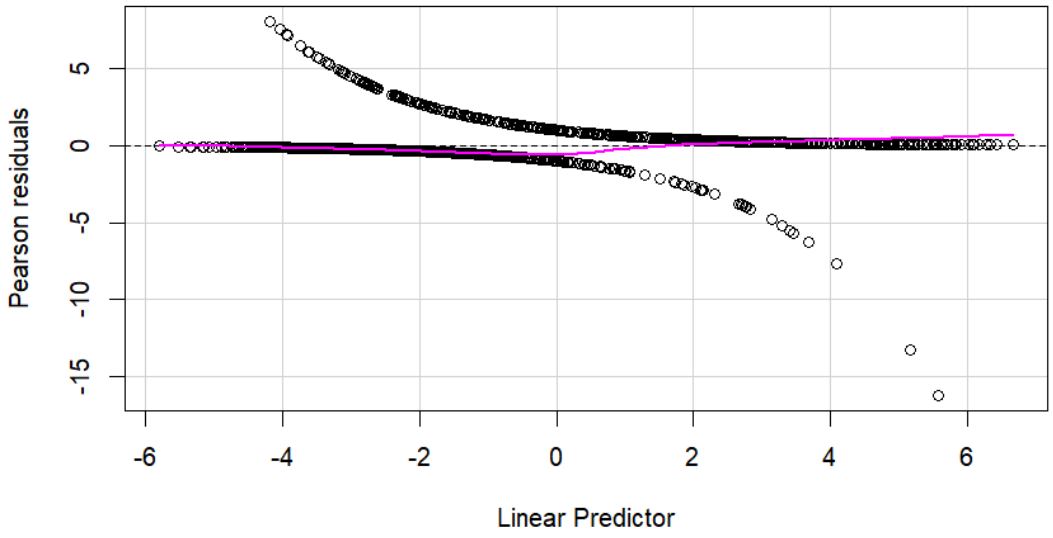

Figura 26: Gràfic de residus de Pearson vs. valors ajustats del model m0.3

Observem que el patró central és força pla al voltant de 0 i no mostra tendències fortes, cosa coherent amb un ajust global raonable del GLM binomial amb link logit. La dispersió dels residus sembla relativament estable al llarg de l'escala dels valors ajustats. Els termes polinòmics milloren lleugerament la linealitat local, però no de manera dramàtica.

També mirem ara si aquest model és millor que el model inicial `m0.1` mitjançant una comparació de AIC i BIC:

```
AIC(m0.1, m0.3)
BIC(m0.1, m0.3)
```

Obtenim els següents resultats:

Model	df	AIC	BIC
m0.1	26	1478.341	1630.128
m0.3	30	1485.137	1648.600

Observem que el model `m0.3` té un AIC i BIC més alts que el model `m0.1`, indicant que no ofereix una millora en l'ajust del model tot i les transformacions aplicades. Per tant, mantenim el model `m0.1` com a model final per a la predicció de l'abandonament.

8.4.2 Model amb dues pendents

Tot i que el model `m0.3` incorporava termes polinòmics per millorar la representació de la relació no lineal amb les variables acadèmiques, els gràfics de residus mostraven encara certes desviacions als extrems. Aquest patró suggeria que la relació entre el nombre d'assignatures aprovades i la probabilitat de *dropout* no és uniforme a tot el rang de valors, sinó que podria canviar a partir d'un determinat punt. Per explorar aquesta hipòtesi, s'ha plantejat un model amb dues pendents, que permet diferenciar el comportament de la relació abans i després d'un llindar concret. Es tracta d'establir un llindar que separi dues regions en la variable acadèmica, on la relació amb la probabilitat de *dropout* pot tenir diferents pendents. Dividim en dos trams les dues variables: cada variable contínua en dos trams (≤ 5 vs > 5 i ≤ 4 vs > 4) i en fem factors binaris.

```
base$Iunits1 <- as.factor(ifelse(base$Curricular_units_1st_sem_approved <= 5, 0, 1))
base$Iunits2a <- as.factor(ifelse(base$Curricular_units_2nd_sem_approved <= 4, 0, 1))
```

Això implica:

- Per al primer semestre, es distingeix entre estudiants amb ≤ 5 assignatures aprovades (nivell baix) i > 5 assignatures aprovades (nivell alt).
- Per al segon semestre, el punt de tall s'ha fixat en 4 assignatures aprovades, observat com a possible canvi de tendència en els gràfics previs.

Aquesta separació és afegida al model com a interaccions amb les variables originals, permetent així que la pendent de la relació pugui variar segons el tram en què es trobi l'estudiant. Ajustem un nou model logístic `m0.5` que incorpora aquestes interaccions:

```
m0.5 <- glm( target ~ Course +
               Mother_s_qualification +
               Father_s_occupation +
               Debtor +
               Tuition_fees_up_to_date +
               Scholarship_holder +
               Age_at_enrollment +
               poly(Curricular_units_1st_sem_approved, 4) * Iunits1 +
               poly(Curricular_units_2nd_sem_approved, 3) * Iunits2a +
               Unemployment_rate,
             data = base,
             family = binomial(link = "logit"))
```

D'aquest model m0.5, realitzem una comparació amb el model inicial `m0.1` i el model `m0.3` mitjançant AIC i BIC, ja que no podem fer un test Òmnibus entre models no imbricats:

```
  AIC(m0.1, m0.3, m0.5)
  BIC(m0.1, m0.3, m0.5)
```

Obtenim els següents resultats:

Model	df	AIC	BIC
m0.1	26	1478.341	1630.128
m0.3	30	1485.137	1648.600
m0.5	32	1584.098	1788.427

Observem que el model `m0.5` té un AIC i BIC més alts que els models `m0.1` i `m0.3`, indicant que no ofereix una millora en l'ajust del model tot i la incorporació de les interaccions amb dues pendents. El model `m0.5`, amb dues pendents i interaccions, permet una millor flexibilitat teòrica en la relació entre el rendiment acadèmic i el risc de *dropout*, però no aporta una millora significativa ni en ajust ni en criteris d'informació respecte als altres models. En conseqüència, es manté el model `m0.1` com a opció òptima per la seva simplicitat i eficiència, ja que descriu adequadament la variabilitat de la resposta amb menys paràmetres i millor penalització segons BIC.

8.4.3 Link probit

Finalment, provem d'ajustar el model final `m0.1` però amb un link probit en lloc del logit per veure si hi ha millores en l'ajust:

```
# Comparació de diferent link
m_probit <- glm(target ~
                  Course +
                  Mother_s_qualification +
                  Father_s_occupation +
                  Debtor +
                  Tuition_fees_up_to_date +
                  Scholarship_holder +
                  Age_at_enrollment +
                  Curricular_units_1st_sem_approved +
                  Curricular_units_2nd_sem_enrolled +
                  Curricular_units_2nd_sem_approved +
                  Unemployment_rate,
                data = base,
                family = binomial(link = "probit"))
```

Comparem els models `m0.1` (logit) i `m_probit` (probit) mitjançant un test Òmnibus:

```
anova(m0.1, m_probit, test = "Chisq")
```

Obtenim els següents resultats:

```
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      2509     1426.3
2      2512     1448.6 -3  -22.272 5.727e-05 ***
```

Plantegem les següents hipòtesis:

- **Hipòtesi nul·la H_0:** No hi ha diferència significativa en l'ajust entre el model logit (`m0.1`) i el model probit (`m_probit`).
- **Hipòtesi alternativa H_1:** Hi ha una diferència significativa en l'ajust entre els dos models.

Com que el p-valor és molt petit (5.727e-05 < 0.05), rebutgem l'hipòtesi nul·la i acceptem l'hipòtesi alternativa. Això indica que hi ha una diferència significativa en l'ajust entre els dos models. Observem que la reducció de la deviància és negativa (-22.272), cosa que indica que el model logit (`m0.1`) té una millor ajust que el model probit (`m_probit`).

També comparem els AIC i BIC dels dos models:

```
AIC(m0.1, m_probit)
BIC(m0.1, m_probit)
```

Obtenim els següents resultats:

Model	df	AIC	BIC
m0.1	26	1478.341	1630.128
m_probit	26	1494.613	1628.886

El model probit té un AIC més alt i un BIC lleugerament més baix, indicant que el model logit ofereix un pitjor ajust a les dades però amb una penalització més baixa per la complexitat del model, però la diferència és minima (de dos punts). En conseqüència, mantenim el model logit `m0.1` com a model final per a la predicció de l'abandonament.

8.5 Corba ROC i AUC

Per avaluar la capacitat predictiva del model final `m0.1`, generem la corba ROC (Receiver Operating Characteristic) i calculem l'AUC (Area Under the Curve). La corba ROC mostra la relació entre la taxa de veritables positius (sensibilitat) i la taxa de falsos positius (1 - especificitat) a diferents llindars de classificació. La AUC quantifica la capacitat del model per distingir entre les dues classes (abandonament sí/no).

Utilitzem la llibreria `pROC` per generar la corba ROC i calcular l'AUC:

```
p_hat <- predict(m0.1, newdata = base, type = "response")

pred <- prediction(p_hat, base$target)
perf <- performance(pred, "tpr", "fpr")  # TPR vs FPR = ROC
plot(perf, col = "#d62728", lwd = 2, main = "ROC - best_mod (ROCR)")
abline(0, 1, lty = 2, col = "grey50")
```

La corba ROC resultant és la següent:

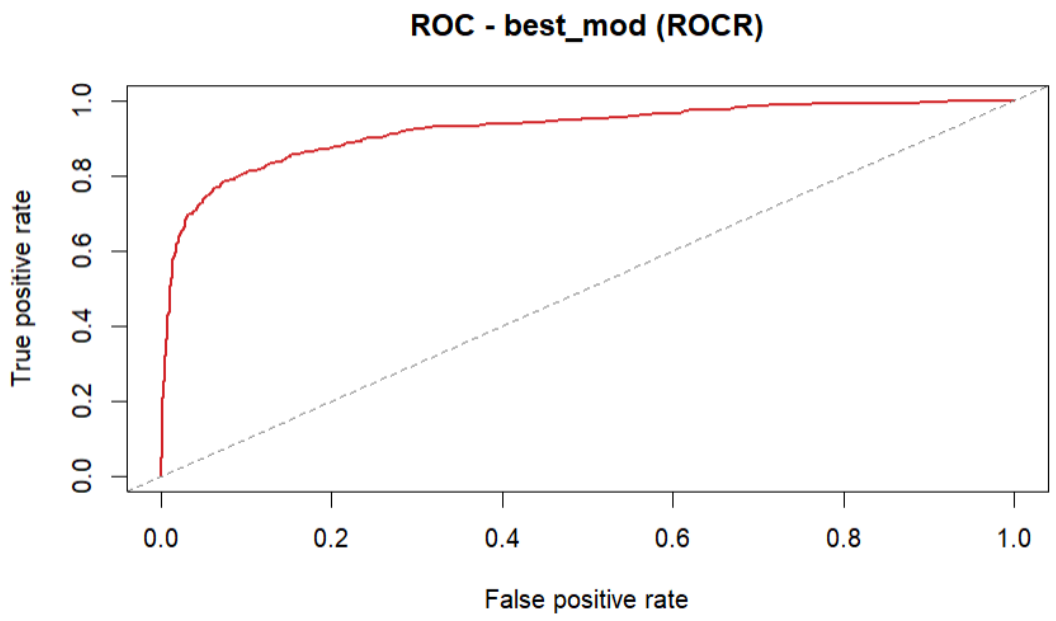

Figura 27: Corba ROC del model final m0.1

La corba ROC mostra una bona capacitat predictiva, ja que es troba per sobre de la línia de referència (diagonal).

Per calcular l'AUC, utilitzem el següent codi:

```
auc_perf <- performance(pred, "auc")
auc_value <- as.numeric(auc_perf@y.values)
auc_value
```

Obtenim un valor d'AUC de `0.9220703`, indicant que el model té una excel·lent capacitat per distingir entre estudiants que abandonen i els que no ho fan. Un AUC proper a 1 suggereix que el model és molt efectiu en la classificació correcta de les observacions.

9. Sèries temporals