432 Class 08

https://thomaselove.github.io/432-2025/

2025-02-06

## Today’s Agenda

* The Bechdel-Wallace Test and the Favorite Movies Data
  + Data Cleanup: Working with Complete Cases
* Logistic Regression with glm() and with lrm()
  + Fitting Five Models and Comparing their Fits
  + Evaluating a Logistic Model in detail
  + Determining an optimal decision rule cutpoint
  + Evaluating Logistic Regression Assumptions
  + Making Predictions and Building Interval Estimates

## Today’s R Setup

knitr::opts\_chunk$set(comment = NA)  
  
library(janitor); library(naniar)  
library(bestglm) ## best subsets search using logistic regression  
library(broom)  
library(car) ## special plot for logistic regression diagnostics  
library(caret) ## for confusion matrices  
library(cobalt) ## new today: to split factor into indicator variables  
library(cutpointr) ## new today: optimizing cutpoints  
library(glue) ## combining R code results and text in labels  
library(gt)  
library(readxl) ## read in data from an Excel file  
library(tableone) ## new today: produce a "simple" Table 1  
library(rms)   
library(easystats); library(tidyverse)  
  
theme\_set(theme\_bw())

# “Favorite Movies” Data

## Ingest Data from an Excel Sheet

mov25\_full <- read\_xlsx("c08/data/movies\_2025-01-28.xlsx", na = c("", "NA"))  
  
dim(mov25\_full)

[1] 228 80

### Select Today’s Variables

mov25\_0 <- mov25\_full |>  
 janitor::clean\_names() |>  
 select(mov\_id, year, mpa, rt\_reviews, ebert, gen\_1,   
 romance, action, bw\_rating, movie) |>  
 mutate(across(where(is.character), as\_factor),  
 mov\_id = as.character(mov\_id),  
 movie = as.character(movie))   
  
dim(mov25\_0)

[1] 228 10

## Variables pulled from mov25\_full

| Variable | Description (n = 228 movies) |
| --- | --- |
| mov\_id | Movie ID (meaningless code) |
| year | Year movie was released |
| mpa | [Motion Picture Association](https://www.motionpictures.org/film-ratings/) rating |
| reviews | Number of Critic Reviews on [rottentomatoes.com](https://www.rottentomatoes.com/) |
| ebert | Star Rating (1-4) on [RogerEbert.com](https://www.rogerebert.com/) |
| gen\_1 | Gender (M or F) of first listed star of film ([IMDB](https://www.imdb.com/)) |
| romance | 1 if Romance is in the movie’s [IMDB](https://www.imdb.com/) Genre list (else 0) |
| action | 1 if Action is in the movie’s [IMDB](https://www.imdb.com/) Genre list (else 0) |
| bw\_rating | Score (0-3) on the [Bechdel-Wallace test](https://bechdeltest.com/) |
| movie | Movie Title |

## The Bechdel Test

The Bechdel Test, or Bechdel-Wallace Test was popularized by Alison Bechdel’s comic, in a 1985 strip called [The Rule](https://dykestowatchoutfor.com/the-rule/).

* from <https://bechdeltest.com/>

The Bechdel-Wallace Test is a simple way to gauge the active presence of female characters in Hollywood films and just how well rounded and complete those roles are[[1]](#footnote-31).

## Passing the Bechdel-Wallace Test

To pass the test, a movie must have all three of the following.

* at least two (named) women
* who talk to each other
* about something besides a man

mov25\_0 <- mov25\_0 |>  
 mutate(bechdel = factor(as.numeric(bw\_rating == 3)))  
mov25\_0 |> tabyl(bechdel, bw\_rating)

bechdel 0 1 2 3 NA\_  
 0 18 61 15 0 0  
 1 0 0 0 124 0  
 <NA> 0 0 0 0 10

I use 0 and 1 as the categories for every logistic regression outcome.

## Some Data Cleanup

1. Drop the films missing the bechdel information.
2. Create an age variable and use it instead of year.

mov25\_0 <- mov25\_0 |>  
 filter(complete.cases(bechdel)) |>  
 mutate(age = 2025-year)  
  
mov25\_0 |> tabyl(bechdel) |> adorn\_pct\_formatting()

bechdel n percent  
 0 94 43.1%  
 1 124 56.9%

Again, I **always** use 0 and 1 for a logistic regression outcome.

## More Data Cleanup

1. Should we collapse the MPA ratings?

summary(mov25\_0$mpa)

PG-13 NR G TV-PG R PG TV-G TV-MA TV-14   
 74 7 7 1 65 61 2 0 1

Let’s collapse to the two largest categories, plus “Other”

mov25\_0 <- mov25\_0 |> mutate(mpa3 = fct\_lump\_n(mpa, n = 2))  
mov25\_0 |> tabyl(mpa3) |> adorn\_pct\_formatting() |>   
 gt() |> tab\_options(table.font.size = 24)

| mpa3 | n | percent |
| --- | --- | --- |
| PG-13 | 74 | 33.9% |
| R | 65 | 29.8% |
| Other | 79 | 36.2% |

## How should we treat ebert rating?

mov25\_0 |> tabyl(ebert) |> adorn\_pct\_formatting()

ebert n percent valid\_percent  
 1.0 4 1.8% 2.0%  
 1.5 3 1.4% 1.5%  
 2.0 19 8.7% 9.7%  
 2.5 20 9.2% 10.2%  
 3.0 47 21.6% 24.0%  
 3.5 43 19.7% 21.9%  
 4.0 60 27.5% 30.6%  
 NA 22 10.1% -

describe(mov25\_0$ebert)

mov25\_0$ebert   
 n missing distinct Info Mean pMedian Gmd   
 196 22 7 0.945 3.204 3.25 0.815   
   
Value 1.0 1.5 2.0 2.5 3.0 3.5 4.0  
Frequency 4 3 19 20 47 43 60  
Proportion 0.020 0.015 0.097 0.102 0.240 0.219 0.306  
  
For the frequency table, variable is rounded to the nearest 0

## What shall we do about missing data?

## using bechdel now instead of bw\_rating  
  
mov25\_0 <- mov25\_0 |>  
 select(mov\_id, bechdel, year, mpa, rt\_reviews, ebert, gen\_1,   
 romance, action, bw\_rating, movie)  
  
miss\_case\_table(mov25\_0)

# A tibble: 2 × 3  
 n\_miss\_in\_case n\_cases pct\_cases  
 <int> <int> <dbl>  
1 0 196 89.9  
2 1 22 10.1

miss\_var\_summary(mov25\_0) |> filter(n\_miss > 0)

# A tibble: 1 × 3  
 variable n\_miss pct\_miss  
 <chr> <int> <num>  
1 ebert 22 10.1

## Which movies are missing data?

mov25\_0 |> filter(!complete.cases(ebert)) |>  
 select(mov\_id, ebert, movie)

# A tibble: 22 × 3  
 mov\_id ebert movie   
 <chr> <dbl> <chr>   
 1 M-001 NA 3 Idiots   
 2 M-031 NA Castle in the Sky (Tenkû no shiro Rapyuta)   
 3 M-038 NA Coming To America   
 4 M-047 NA Dilwale Dulhania Le Jayenge (The Brave Heart Will Take the Brid…  
 5 M-060 NA A Fistful of Dollars   
 6 M-087 NA High School Musical 2   
 7 M-088 NA The Hobbit: An Unexpected Journey   
 8 M-091 NA Hot Fuzz   
 9 M-111 NA Kiki's Delivery Service (Majo no takkyûbin)   
10 M-130 NA Madea Goes To Jail   
# ℹ 12 more rows

* Does imputing seem reasonable here?
* Can we assume Missing at Random?

## Today, we’ll use complete cases

* Here is my recipe for mov25 from start to finish.

mov25 <- read\_xlsx("c08/data/movies\_2025-01-28.xlsx", na = c("", "NA")) |>  
 janitor::clean\_names() |>  
 filter(complete.cases(bw\_rating, ebert)) |>  
 mutate(across(where(is.character), as\_factor),  
 mov\_id = as.character(mov\_id),  
 movie = as.character(movie),  
 age = 2025-year,  
 mpa3 = fct\_lump\_n(mpa, n = 2),  
 bechdel = factor(as.numeric(bw\_rating == 3))) |>  
 select(mov\_id, bechdel, age, mpa3, reviews = rt\_reviews, ebert,   
 gen\_1, romance, action, movie)  
  
dim(mov25)

[1] 196 10

## mov25 Variable Descriptions

| Variable | Description (n = 196 movies) |
| --- | --- |
| mov\_id | Movie ID (meaningless code) |
| bechdel | 1 (“Pass”) or 0 (“Fail”) the [Bechdel-Wallace test](https://bechdeltest.com/) |
| age | Age of movie (2025 - Year of release) |
| mpa3 | [MPA](https://www.motionpictures.org/film-ratings/) rating (3 levels: PG-13, R, Other) |
| reviews | Number of Critic Reviews on [rottentomatoes.com](https://www.rottentomatoes.com/) |
| ebert | Star Rating (1-4) on [RogerEbert.com](https://www.rogerebert.com/) |
| gen\_1 | Gender (M or F) of first listed star of film ([IMDB](https://www.imdb.com/)) |
| romance | 1 if Romance is in the movie’s [IMDB](https://www.imdb.com/) Genre list (else 0) |
| action | 1 if Action is in the movie’s [IMDB](https://www.imdb.com/) Genre list (else 0) |
| movie | Movie Title |

## data\_codebook()

data\_codebook(mov25 |> select(-mov\_id, -movie))

select(mov25, -mov\_id, -movie) (196 rows and 8 variables, 8 shown)  
  
ID | Name | Type | Missings | Values | N  
---+---------+-------------+----------+-----------+------------  
1 | bechdel | categorical | 0 (0.0%) | 0 | 83 (42.3%)  
 | | | | 1 | 113 (57.7%)  
---+---------+-------------+----------+-----------+------------  
2 | age | numeric | 0 (0.0%) | [1, 83] | 196  
---+---------+-------------+----------+-----------+------------  
3 | mpa3 | categorical | 0 (0.0%) | PG-13 | 67 (34.2%)  
 | | | | R | 61 (31.1%)  
 | | | | Other | 68 (34.7%)  
---+---------+-------------+----------+-----------+------------  
4 | reviews | numeric | 0 (0.0%) | [25, 602] | 196  
---+---------+-------------+----------+-----------+------------  
5 | ebert | numeric | 0 (0.0%) | [1, 4] | 196  
---+---------+-------------+----------+-----------+------------  
6 | gen\_1 | categorical | 0 (0.0%) | M | 149 (76.0%)  
 | | | | F | 47 (24.0%)  
---+---------+-------------+----------+-----------+------------  
7 | romance | numeric | 0 (0.0%) | 0 | 151 (77.0%)  
 | | | | 1 | 45 (23.0%)  
---+---------+-------------+----------+-----------+------------  
8 | action | numeric | 0 (0.0%) | 0 | 142 (72.4%)  
 | | | | 1 | 54 (27.6%)  
---------------------------------------------------------------

## The mov25 data listing

mov25

# A tibble: 196 × 10  
 mov\_id bechdel age mpa3 reviews ebert gen\_1 romance action movie   
 <chr> <fct> <dbl> <fct> <dbl> <dbl> <fct> <dbl> <dbl> <chr>   
 1 M-002 1 62 Other 60 4 M 0 0 8 1/2   
 2 M-003 1 26 PG-13 92 2.5 M 1 0 10 Things I Ha…  
 3 M-004 0 57 Other 118 4 M 0 0 2001: A Space …  
 4 M-005 1 16 Other 72 4 F 0 0 About Elly (Da…  
 5 M-006 1 12 R 168 2.5 M 1 0 About Time   
 6 M-007 1 46 R 135 4 F 0 0 Alien   
 7 M-008 1 41 Other 154 4 M 0 0 Amadeus   
 8 M-009 1 16 PG-13 335 4 M 0 1 Avatar   
 9 M-010 1 7 PG-13 492 2.5 M 0 1 Avengers: Infi…  
10 M-011 1 6 PG-13 556 3 M 0 1 Avengers: Endg…  
# ℹ 186 more rows

identical(nrow(mov25), n\_distinct(mov25$mov\_id)) ## are IDs unique?

[1] TRUE

## Splitting the sample?

We have 196 films in our mov25 tibble.

* It turns out that a logistic regression model needs about 96 observations just to fit a reasonable intercept term.
* Each additional coefficient we fit requires another 10-20 observations just so that we *might* validate well.

Here, we want to explore seven predictors (age, mpa3, reviews, ebert, gen\_1, romance and action.)

* Does it make sense to split our data into separate training and testing samples?

## Four of Dr. Love’s 125 favorite movies

love4 <- tibble(  
 mov\_id = c("L-2", "L-8", "L-63", "L-125"), age = c(63, 37, 22, 30),   
 bechdel = c(0, 0, 1, 1), gen\_1 = c("M", "M", "M", "F"),   
 mpa3 = c("PG-13", "R", "Other", "Other"), reviews = c(67, 89, 230, 67),   
 ebert = c(4, 2, 3.5, 3.5), romance = c(0, 0, 1, 1), action = c(0, 1, 0, 0),   
 movie = c("The Manchurian Candidate", "Die Hard", "Love Actually",   
 "Sense and Sensibility")) |>  
 mutate(across(where(is.character), as\_factor),  
 mov\_id = as.character(mov\_id),  
 movie = as.character(movie))   
love4

# A tibble: 4 × 10  
 mov\_id age bechdel gen\_1 mpa3 reviews ebert romance action movie   
 <chr> <dbl> <dbl> <fct> <fct> <dbl> <dbl> <dbl> <dbl> <chr>   
1 L-2 63 0 M PG-13 67 4 0 0 The Manchurian …  
2 L-8 37 0 M R 89 2 0 1 Die Hard   
3 L-63 22 1 M Other 230 3.5 1 0 Love Actually   
4 L-125 30 1 F Other 67 3.5 1 0 Sense and Sensi…

## The Logistic Regression Model

Here, our *event* will be “movie passes the Bechdel-Wallace test” (bechdel = 1)

## Guess the associations?

In which direction do you think these variables will be associated with passing the Bechdel-Wallace test?

| Variable | Description (n = 196 movies) |
| --- | --- |
| age | Age of movie (2025 - Year of release) |
| mpa3 | [MPA](https://www.motionpictures.org/film-ratings/) rating (3 levels: PG-13, R, Other) |
| reviews | Number of Critic Reviews on [rottentomatoes.com](https://www.rottentomatoes.com/) |
| ebert | Star Rating (1-4) on [RogerEbert.com](https://www.rogerebert.com/) |
| gen\_1 | Gender (M or F) of first listed star of film ([IMDB](https://www.imdb.com/)) |
| romance | 1 if Romance is in the movie’s [IMDB](https://www.imdb.com/) Genre list (else 0) |
| action | 1 if Action is in the movie’s [IMDB](https://www.imdb.com/) Genre list (else 0) |

## Table 1?

CreateTableOne(data = mov25,   
 vars = c("age", "mpa3", "reviews", "ebert", "gen\_1", "romance", "action"),  
 factorVars = c("mpa3", "gen\_1", "romance", "action"),  
 strata = "bechdel")

Stratified by bechdel  
 0 1 p test  
 n 83 113   
 age (mean (SD)) 24.42 (15.50) 20.05 (13.13) 0.034   
 mpa3 (%) 0.387   
 PG-13 25 (30.1) 42 (37.2)   
 R 30 (36.1) 31 (27.4)   
 Other 28 (33.7) 40 (35.4)   
 reviews (mean (SD)) 183.84 (91.46) 227.20 (135.10) 0.012   
 ebert (mean (SD)) 3.33 (0.66) 3.11 (0.80) 0.041   
 gen\_1 = F (%) 3 ( 3.6) 44 (38.9) <0.001   
 romance = 1 (%) 11 (13.3) 34 (30.1) 0.009   
 action = 1 (%) 29 (34.9) 25 (22.1) 0.068

## What Five Models Will We Fit Today?

* fit1: two predictors (age, gen\_1)
* fit2: all seven predictors (age, gen\_1, mpa3, reviews, ebert, romance, action)
* fit3: predictor subset with the lowest BIC (from bestglm search)
* fit4: predictor subset with the lowest AIC (bestglm)
* fit5: add two non-linear terms using our 7 predictors

# Model fit1

## Model fit1: two predictors

fit1 <- glm(bechdel ~ age + gen\_1,  
 data = mov25, family = binomial(link = "logit"))  
  
n\_obs(fit1)

[1] 196

performance\_roc(fit1)

AUC: 71.99%

model\_performance(fit1)

# Indices of model performance  
  
AIC | AICc | BIC | Tjur's R2 | RMSE | Sigma | Log\_loss | Score\_log  
------------------------------------------------------------------------------  
231.153 | 231.278 | 240.987 | 0.181 | 0.447 | 1.000 | 0.574 | -126.197  
  
AIC | Score\_spherical | PCP  
---------------------------------  
231.153 | 0.005 | 0.600

## Model fit1 parameters

model\_parameters(fit1, exponentiate = TRUE, ci = 0.90)

Parameter | Odds Ratio | SE | 90% CI | z | p  
-----------------------------------------------------------------  
(Intercept) | 1.33 | 0.41 | [0.81, 2.21] | 0.94 | 0.348   
age | 0.98 | 0.01 | [0.96, 1.00] | -1.67 | 0.095   
gen 1 [F] | 16.55 | 10.28 | [6.64, 53.82] | 4.52 | < .001

Uncertainty intervals (profile-likelihood) and p-values (two-tailed)  
 computed using a Wald z-distribution approximation.

Some coefficients seem to be rather large, which may indicate issues  
 with (quasi) complete separation. Consider using bias-corrected or  
 penalized regression models.

or, if you prefer…

tidy(fit1, exponentiate = TRUE, conf.int = TRUE, conf.level = 0.90)

# A tibble: 3 × 7  
 term estimate std.error statistic p.value conf.low conf.high  
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 (Intercept) 1.33 0.305 0.938 0.348 0.808 2.21   
2 age 0.981 0.0116 -1.67 0.0948 0.962 0.999  
3 gen\_1F 16.6 0.621 4.52 0.00000619 6.64 53.8

## What is the fit1 equation?

fit1$coefficients ## note: without exponentiation

(Intercept) age gen\_1F   
 0.28623530 -0.01932756 2.80648551

$$
logit(\mbox{bechdel = 1}) = \\
log\left( \frac{Pr(\mbox{bechdel = 1})}{1 - Pr(\mbox{bechdel = 1})} \right) = \\
0.2862 -0.0193 (\mbox{age}) + 2.8065 (\mbox{gen\_1 = F})
$$

## lrm version of fit1

d <- datadist(mov25); options(datadist = "d")  
  
fit1\_lrm <- lrm(bechdel ~ age + gen\_1, data = mov25,   
 x = TRUE, y = TRUE)

|  |
| --- |
| Key Summaries for fit1\_lrm include… |
| * C = 0.720, Nagelkerke = 0.259, Brier score = 0.200 * See next slide for details. |

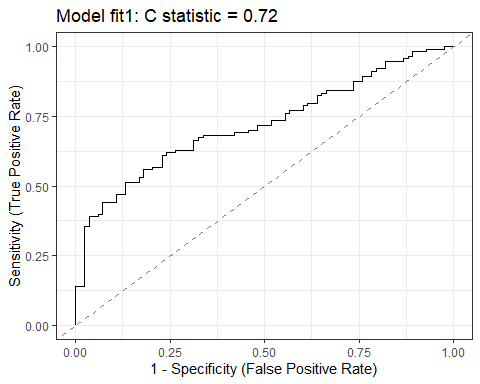
## fit1\_lrm summaries

fit1\_lrm

Logistic Regression Model  
  
lrm(formula = bechdel ~ age + gen\_1, data = mov25, x = TRUE,   
 y = TRUE)  
  
 Model Likelihood Discrimination Rank Discrim.   
 Ratio Test Indexes Indexes   
Obs 196 LR chi2 41.95 R2 0.259 C 0.720   
 0 83 d.f. 2 R2(2,196)0.184 Dxy 0.440   
 1 113 Pr(> chi2) <0.0001 R2(2,143.6)0.243 gamma 0.446   
max |deriv| 3e-06 Brier 0.200 tau-a 0.216   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept 0.2862 0.3051 0.94 0.3482   
age -0.0193 0.0116 -1.67 0.0948   
gen\_1=F 2.8065 0.6209 4.52 <0.0001

## ROC Curve Analysis for fit1

plot(performance\_roc(fit1)) +  
 labs(title = glue("Model fit1: C statistic = ",  
 round\_half\_up(as.numeric(performance\_roc(fit1)),3)))



## Nagelkerke for fit1

From the lrm() fit (fit1\_lrm), we have: Nagelkerke = 0.259, which isn’t 25.9% of anything, but will be 1 if the fitted model shows as much improvement as possible over the null model.

* This is the version of that the validate() function will look at.
* The validated version of this is the best way I have to compare models.

## Tjur’s for fit1

From the glm() fit (fit1), and the model\_performance() function, we have: Tjur’s = 0.181.

* Tjur’s = the difference between the average fitted probability when the outcome is 1 (bechdel Pass) minus the average fitted probability when the outcome is 0 (bechdel Fail, here.)
* It’s less useful than a validated Nagelkerke if your goal is to compare two models.

## The McFadden for fit1

* The McFadden , which is 1 minus the ratio of (the model deviance over the deviance for the null model.)

fit1\_lrm$deviance

[1] 267.1038 225.1527

1 - (fit1\_lrm$deviance[2] / fit1\_lrm$deviance[1])

[1] 0.157059

The McFadden (= 0.157 here) and approximates a proportionate reduction in error.

* Also less useful than a validated Nagelkerke if your goal is to compare two models.

## Key fit1 Pseudo- measures

r2\_fit1 <- tibble( name = c("McFadden", "Nagelkerke", "Tjur"),  
 r\_sq = c(as.numeric(r2\_mcfadden(fit1)[1]),  
 as.numeric(r2\_nagelkerke(fit1)),  
 as.numeric(r2\_tjur(fit1))))  
  
r2\_fit1 |> gt() |> fmt\_number(decimals = 4) |>   
 tab\_options(table.font.size = 24) |>  
 opt\_stylize(style = 2, color = "pink")

| name | r\_sq |
| --- | --- |
| McFadden | 0.1571 |
| Nagelkerke | 0.2590 |
| Tjur | 0.1813 |

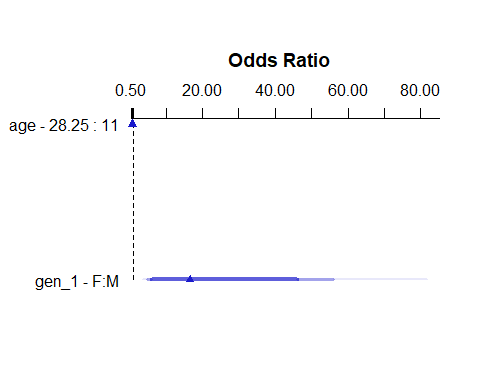
## What are the effect sizes in fit1\_lrm?

summary(fit1\_lrm)

Effects Response : bechdel   
  
 Factor Low High Diff. Effect S.E. Lower 0.95 Upper 0.95  
 age 11 28.25 17.25 -0.33340 0.19955 -0.72451 0.057713   
 Odds Ratio 11 28.25 17.25 0.71648 NA 0.48456 1.059400   
 gen\_1 - F:M 1 2.00 NA 2.80650 0.62091 1.58950 4.023500   
 Odds Ratio 1 2.00 NA 16.55200 NA 4.90140 55.894000

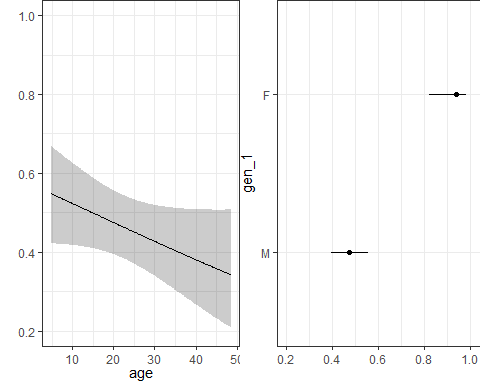
## Plotting the Effects for fit1\_lrm

plot(summary(fit1\_lrm))



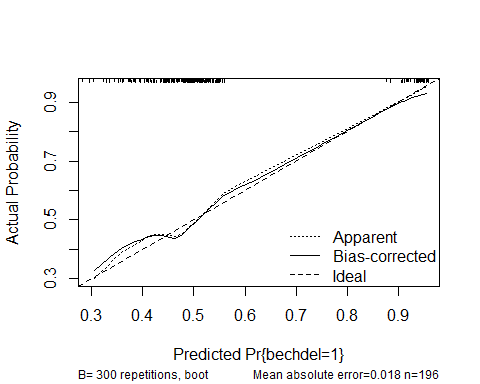
## Prediction Plots for fit1\_lrm

ggplot(Predict(fit1\_lrm, fun = plogis), layout = c(1,2))



## Calibration Plot for fit1\_lrm

set.seed(4321231)  
plot(calibrate(fit1\_lrm, method = "boot", B = 300))



n=196 Mean absolute error=0.018 Mean squared error=0.00046  
0.9 Quantile of absolute error=0.036

## fit1 Hosmer-Lemeshow test

* This is a somewhat out-of-date method for assessing calibration. Split the data into (here) 5 bins (default is 10) and look for a test result that has a large value.

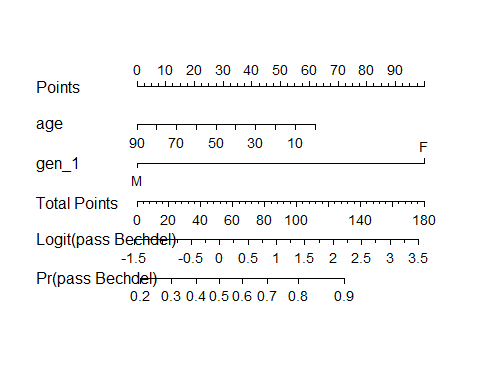
performance\_hosmer(fit1, n\_bins = 5)

# Hosmer-Lemeshow Goodness-of-Fit Test  
  
 Chi-squared: 1.224  
 df: 3   
 p-value: 0.747

Summary: model seems to fit well.

## Nomogram for fit1\_lrm

plot(nomogram(fit1\_lrm, fun = plogis, funlabel = "Pr(pass Bechdel)"),  
 lplabel = "Logit(pass Bechdel)")



## Predictions from fit1 for love4 movies

augment(fit1, newdata = love4, type.predict = "response") |>  
 select(movie, .fitted, bechdel, everything()) |>  
 gt() |> tab\_options(table.font.size = 20) |>  
 opt\_stylize(style = 5, color = "pink")

| movie | .fitted | bechdel | mov\_id | age | gen\_1 | mpa3 | reviews | ebert | romance | action |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The Manchurian Candidate | 0.2826406 | 0 | L-2 | 63 | M | PG-13 | 67 | 4.0 | 0 | 0 |
| Die Hard | 0.3943927 | 0 | L-8 | 37 | M | R | 89 | 2.0 | 0 | 1 |
| Love Actually | 0.4653130 | 1 | L-63 | 22 | M | Other | 230 | 3.5 | 1 | 0 |
| Sense and Sensibility | 0.9250408 | 1 | L-125 | 30 | F | Other | 67 | 3.5 | 1 | 0 |

## CIs around our predictions?

augment(fit1, newdata = love4, type.predict = "link", se\_fit = TRUE) |>  
 mutate(ci\_90\_low = .fitted - 1.645 \* .se.fit,   
 ci\_90\_high = .fitted + 1.645 \* .se.fit) |>  
 select(movie, .fitted, .se.fit, ci\_90\_low, ci\_90\_high, bechdel, everything())

# A tibble: 4 × 14  
 movie .fitted .se.fit ci\_90\_low ci\_90\_high bechdel mov\_id age gen\_1 mpa3   
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <fct> <fct>  
1 The Man… -0.931 0.501 -1.76 -0.107 0 L-2 63 M PG-13  
2 Die Hard -0.429 0.239 -0.822 -0.0361 0 L-8 37 M R   
3 Love Ac… -0.139 0.166 -0.412 0.134 1 L-63 22 M Other  
4 Sense a… 2.51 0.604 1.52 3.51 1 L-125 30 F Other  
# ℹ 4 more variables: reviews <dbl>, ebert <dbl>, romance <dbl>, action <dbl>

## Converting from Logit to Probability Scale

For The Manchurian Candidate, our predicted logit(bechdel pass) = -0.9314, with 90% CI (-1.7553, -0.1075).

* If logit(bechdel pass) = -0.9314, then odds(bechdel pass) = exp(-0.9314), and pr(bechdel pass) = exp(-0.9314) / (1 + exp(-0.9314)) = 0.3940 / 1.3940 = 0.283
* If logit(bechdel pass) = -1.7553, then odds(bechdel pass) = exp(-1.7553), and pr(bechdel pass) = exp(-1.7553) / (1 + exp(-1.7553)) = 0.1729 / 1.1729 = 0.147
* If logit(bechdel pass) = -0.1075, then odds(bechdel pass) = exp(-0.1075), and pr(bechdel pass) = exp(-0.1075) / (1 + exp(-0.1075)) = 0.8981 / 1.8981 = 0.473

Predicted prob(bechdel pass) = 0.283 with 90% confidence interval (0.147, 0.473) for The Manchurian Candidate using fit1.

## Confusion Matrix: Picking a Decision Rule

* We’ll use cutpointr to select a decision rule which maximizes “Sensitivity” + “Specificity”.

fit1\_aug <- augment(fit1, type.predict = "response")  
  
cp1 <- cutpointr(data = fit1\_aug, .fitted, bechdel,   
 pos\_class = 1, neg\_class = 0,  
 method = maximize\_metric, metric = sum\_sens\_spec)

Assuming the positive class has higher x values

cp1 |> select(direction, optimal\_cutpoint, method, sum\_sens\_spec) |>   
 gt() |> tab\_options(table.font.size = 24) |>   
 opt\_stylize(style = 2, color = "pink")

| direction | optimal\_cutpoint | method | sum\_sens\_spec |
| --- | --- | --- | --- |
| >= | 0.5328563 | maximize\_metric | 1.380744 |

## Confusion Matrix for fit1

cm1 <- confusionMatrix(data = factor(fit1\_aug$.fitted >= cp1$optimal\_cutpoint),  
 reference = factor(fit1\_aug$bechdel == 1), positive = "TRUE")  
cm1

Confusion Matrix and Statistics  
  
 Reference  
Prediction FALSE TRUE  
 FALSE 72 55  
 TRUE 11 58  
   
 Accuracy : 0.6633   
 95% CI : (0.5925, 0.729)  
 No Information Rate : 0.5765   
 P-Value [Acc > NIR] : 0.008018   
   
 Kappa : 0.3557   
   
 Mcnemar's Test P-Value : 1.204e-07   
   
 Sensitivity : 0.5133   
 Specificity : 0.8675   
 Pos Pred Value : 0.8406   
 Neg Pred Value : 0.5669   
 Prevalence : 0.5765   
 Detection Rate : 0.2959   
 Detection Prevalence : 0.3520   
 Balanced Accuracy : 0.6904   
   
 'Positive' Class : TRUE

## Percentage of Correct Predictions

performance\_pcp(fit1, ci = 0.90, method = "Herron")

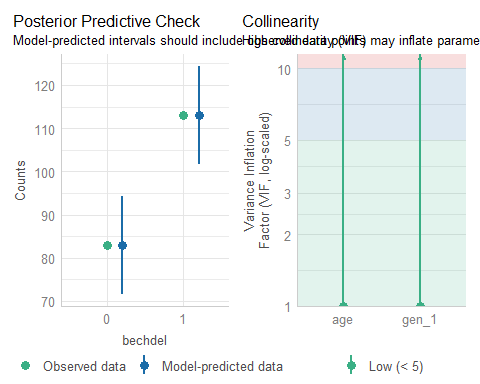
# Percentage of Correct Predictions from Logistic Regression Model  
  
 Full model: 60.02% [54.27% - 65.78%]  
 Null model: 51.17% [45.30% - 57.04%]  
  
# Likelihood-Ratio-Test  
  
 Chi-squared: 41.951  
 df: 2.000  
 p-value: 0.000

* Herron’s PCP = sum of predicted probabilities where y=1, plus the sum of (1 - predicted probabilities) where y=0, divided by the number of observations.

## check\_model for fit1 (1/4)

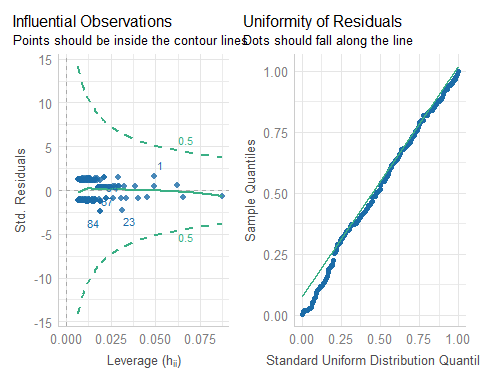
check\_model(fit1, check = c("pp\_check", "vif"))

Cannot simulate residuals for models of class `glm`. Please try  
 `check\_model(..., residual\_type = "normal")` instead.



## check\_model for fit1 (2/4)

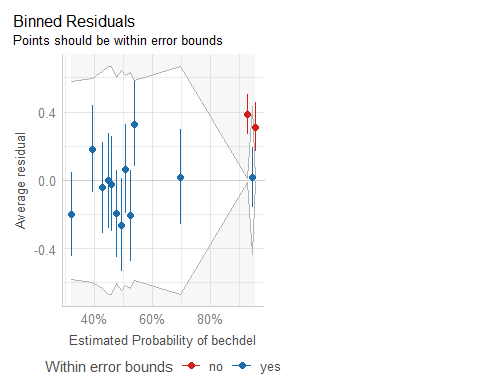
check\_model(fit1, check = c("outliers", "qq"))



## check\_model for fit1 (3/4)

check\_model(fit1, check = c("binned\_residuals"))

Cannot simulate residuals for models of class `glm`. Please try  
 `check\_model(..., residual\_type = "normal")` instead.



## check\_model for fit1 (4/4)

* Extra details for the last three plots…

check\_outliers(fit1)

OK: No outliers detected.  
- Based on the following method and threshold: cook (0.5).  
- For variable: (Whole model)

check\_residuals(fit1)

OK: Simulated residuals appear as uniformly distributed (p = 0.347).

binned\_residuals(fit1)

Warning: About 86% of the residuals are inside the error bounds (~95% or higher would be good).

## Analysis of Deviance for fit1

anova(fit1\_lrm)

Wald Statistics Response: bechdel   
  
 Factor Chi-Square d.f. P   
 age 2.79 1 0.0948  
 gen\_1 20.43 1 <.0001  
 TOTAL 23.02 2 <.0001

|  |
| --- |
| Note |
| Remember that this result shows sequential tests, and if you change the order of the predictors, the *p* values will change. |

## Validating Key Summaries (fit1)

set.seed(202502061); validate(fit1\_lrm, B = 300)

index.orig training test optimism index.corrected n  
Dxy 0.4401 0.4350 0.4314 0.0035 0.4366 300  
R2 0.2590 0.2659 0.2539 0.0120 0.2469 300  
Intercept 0.0000 0.0000 -0.0159 0.0159 -0.0159 300  
Slope 1.0000 1.0000 0.9404 0.0596 0.9404 300  
Emax 0.0000 0.0000 0.0162 0.0162 0.0162 300  
D 0.2089 0.2166 0.2043 0.0124 0.1966 300  
U -0.0102 -0.0102 0.0229 -0.0331 0.0229 300  
Q 0.2191 0.2268 0.1814 0.0454 0.1737 300  
B 0.1998 0.1980 0.2023 -0.0044 0.2042 300  
g 1.2422 1.5587 1.1993 0.3593 0.8829 300  
gp 0.2164 0.2128 0.2093 0.0035 0.2129 300

* C = 0.5 + Dxy, so validated C for fit1 = 0.5 + (0.4366/2) = 0.7183, validated Nagelkerke = 0.2469, and validated Brier score B = 0.2042

## Cross-Validating fit1 AUC

If we wanted to obtain a cross-validated (with 5 folds) C statistic (AUC) from a glm() fit, we can do so with performance\_accuracy()…

set.seed(123451)  
performance\_accuracy(fit1, method = "cv", k = 5, ci = 0.90)

# Accuracy of Model Predictions  
  
Accuracy (90% CI): 72.93% [56.22%, 84.25%]  
Method: Area under Curve

# Model fit2

## Model fit2: all seven predictors

fit2 <- glm(bechdel ~ age + gen\_1 + mpa3 + reviews +   
 ebert + romance + action,  
 data = mov25, family = binomial(link = "logit"))  
  
n\_obs(fit2)

[1] 196

performance\_roc(fit2)

AUC: 79.21%

model\_performance(fit2)

# Indices of model performance  
  
AIC | AICc | BIC | Tjur's R2 | RMSE | Sigma | Log\_loss | Score\_log  
------------------------------------------------------------------------------  
227.984 | 228.952 | 257.487 | 0.255 | 0.426 | 1.000 | 0.536 | -135.916  
  
AIC | Score\_spherical | PCP  
---------------------------------  
227.984 | 0.005 | 0.636

## Model fit2 parameters

model\_parameters(fit2, exponentiate = TRUE, ci = 0.90)

Parameter | Odds Ratio | SE | 90% CI | z | p  
---------------------------------------------------------------------  
(Intercept) | 0.95 | 0.90 | [0.20, 4.53] | -0.06 | 0.955   
age | 1.02 | 0.02 | [0.99, 1.05] | 0.97 | 0.332   
gen 1 [F] | 13.28 | 8.50 | [5.13, 44.26] | 4.04 | < .001  
mpa3 [R] | 0.74 | 0.34 | [0.35, 1.56] | -0.66 | 0.512   
mpa3 [Other] | 1.11 | 0.50 | [0.52, 2.35] | 0.23 | 0.821   
reviews | 1.01 | 2.35e-03 | [1.00, 1.01] | 3.06 | 0.002   
ebert | 0.59 | 0.16 | [0.37, 0.91] | -1.94 | 0.053   
romance | 1.69 | 0.79 | [0.79, 3.71] | 1.11 | 0.265   
action | 0.46 | 0.20 | [0.23, 0.93] | -1.79 | 0.074

Uncertainty intervals (profile-likelihood) and p-values (two-tailed)  
 computed using a Wald z-distribution approximation.

## What is the fit2 equation?

fit2$coefficients ## note: without exponentiation

(Intercept) age gen\_1F mpa3R mpa3Other reviews   
-0.053062946 0.016521393 2.586249907 -0.296613775 0.103018693 0.007152862   
 ebert romance action   
-0.528343279 0.522336854 -0.766365711

$$
logit(\mbox{bechdel = 1}) = \\
-0.0531 + 0.0165 (\mbox{age}) + 2.5862 (\mbox{gen\_1 = F}) - 0.2966 (\mbox{mpa3 = R}) + \\
0.1030 (\mbox{mpa3 = Other}) + 0.0072 (\mbox{reviews}) - 0.5283 (\mbox{ebert}) + \\
0.5223 (\mbox{romance}) - 0.7663 (\mbox{action})
$$

## lrm version of fit2

d <- datadist(mov25); options(datadist = "d")  
  
fit2\_lrm <- lrm(bechdel ~ age + gen\_1 + mpa3 + reviews +   
 ebert + romance + action,   
 data = mov25, x = TRUE, y = TRUE)

|  |
| --- |
| Key Summaries for fit2\_lrm include… |
| * C = 0.792, Nagelkerke = 0.340, Brier score = 0.181 * See next slide for details. |

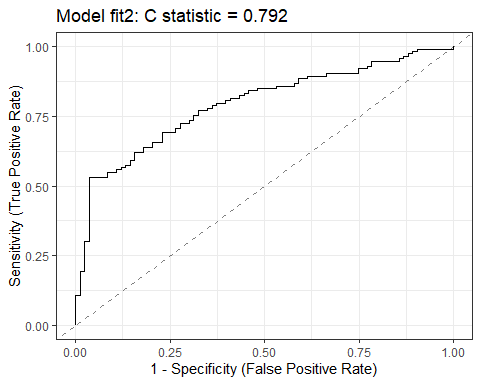
## fit2\_lrm summaries

fit2\_lrm

Logistic Regression Model  
  
lrm(formula = bechdel ~ age + gen\_1 + mpa3 + reviews + ebert +   
 romance + action, data = mov25, x = TRUE, y = TRUE)  
  
 Model Likelihood Discrimination Rank Discrim.   
 Ratio Test Indexes Indexes   
Obs 196 LR chi2 57.12 R2 0.340 C 0.792   
 0 83 d.f. 8 R2(8,196)0.222 Dxy 0.584   
 1 113 Pr(> chi2) <0.0001 R2(8,143.6)0.290 gamma 0.584   
max |deriv| 8e-05 Brier 0.181 tau-a 0.287   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept -0.0531 0.9450 -0.06 0.9552   
age 0.0165 0.0170 0.97 0.3316   
gen\_1=F 2.5862 0.6403 4.04 <0.0001   
mpa3=R -0.2966 0.4518 -0.66 0.5115   
mpa3=Other 0.1030 0.4550 0.23 0.8209   
reviews 0.0072 0.0023 3.06 0.0022   
ebert -0.5283 0.2727 -1.94 0.0527   
romance 0.5223 0.4689 1.11 0.2653   
action -0.7664 0.4290 -1.79 0.0741

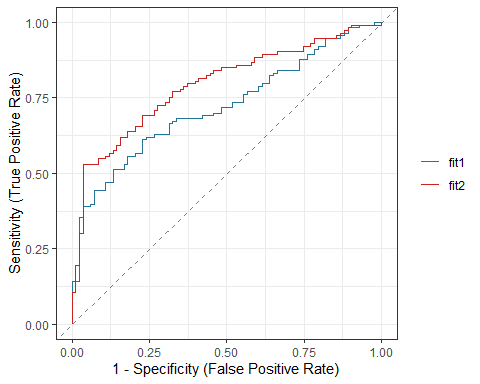
## ROC Curve Analysis for fit2

plot(performance\_roc(fit2)) +  
 labs(title = glue("Model fit2: C statistic = ",  
 round\_half\_up(as.numeric(performance\_roc(fit2)),3)))



## Comparing fit1 to fit2 (ROC)

plot(performance\_roc(fit1, fit2)) +  
 scale\_color\_social()



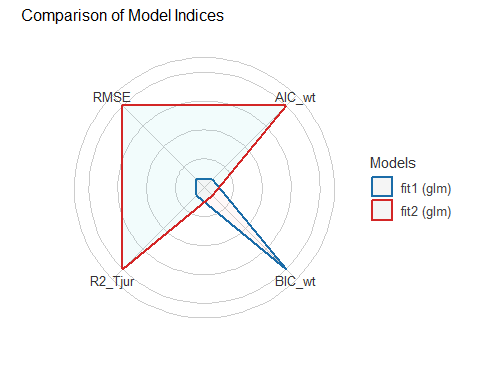
## Pseudo- measures (models so far)

r2\_res12 <- tibble(name = c("fit1", "fit2"),   
 McFadden = c(as.numeric(r2\_mcfadden(fit1)[1]),  
 as.numeric(r2\_mcfadden(fit2)[1])),  
 Nagelkerke = c(as.numeric(r2\_nagelkerke(fit1)),  
 as.numeric(r2\_nagelkerke(fit2))),  
 Tjur = c(as.numeric(r2\_tjur(fit1)),  
 as.numeric(r2\_tjur(fit2))))  
  
r2\_res12 |> gt() |> fmt\_number(decimals = 4) |>   
 tab\_options(table.font.size = 24) |>  
 opt\_stylize(style = 4, color = "green")

| name | McFadden | Nagelkerke | Tjur |
| --- | --- | --- | --- |
| fit1 | 0.1571 | 0.2590 | 0.1813 |
| fit2 | 0.2138 | 0.3398 | 0.2549 |

## Comparing Model Indices from fit1 and fit2

plot(compare\_performance(fit1, fit2, metrics = "common"))



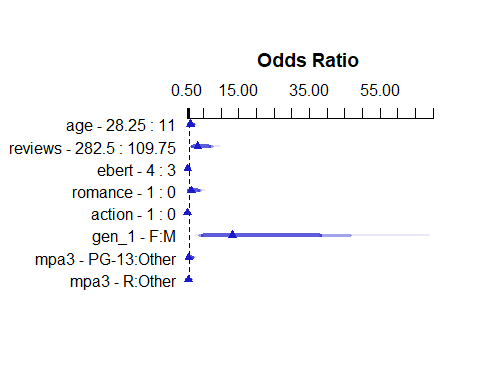
## What are the effect sizes in fit2\_lrm?

summary(fit2\_lrm)

Effects Response : bechdel   
  
 Factor Low High Diff. Effect S.E. Lower 0.95 Upper 0.95  
 age 11.00 28.25 17.25 0.28499 0.29355 -0.29035 0.8603300  
 Odds Ratio 11.00 28.25 17.25 1.32980 NA 0.74800 2.3640000  
 reviews 109.75 282.50 172.75 1.23570 0.40358 0.44465 2.0267000  
 Odds Ratio 109.75 282.50 172.75 3.44060 NA 1.55990 7.5887000  
 ebert 3.00 4.00 1.00 -0.52834 0.27268 -1.06280 0.0060977  
 Odds Ratio 3.00 4.00 1.00 0.58958 NA 0.34549 1.0061000  
 romance 0.00 1.00 1.00 0.52234 0.46894 -0.39677 1.4414000  
 Odds Ratio 0.00 1.00 1.00 1.68600 NA 0.67249 4.2268000  
 action 0.00 1.00 1.00 -0.76637 0.42903 -1.60720 0.0745170  
 Odds Ratio 0.00 1.00 1.00 0.46470 NA 0.20044 1.0774000  
 gen\_1 - F:M 1.00 2.00 NA 2.58620 0.64033 1.33120 3.8413000  
 Odds Ratio 1.00 2.00 NA 13.28000 NA 3.78570 46.5840000  
 mpa3 - PG-13:Other 3.00 1.00 NA -0.10302 0.45498 -0.99475 0.7887200  
 Odds Ratio 3.00 1.00 NA 0.90211 NA 0.36981 2.2006000  
 mpa3 - R:Other 3.00 2.00 NA -0.39963 0.42825 -1.23900 0.4397200  
 Odds Ratio 3.00 2.00 NA 0.67057 NA 0.28968 1.5523000

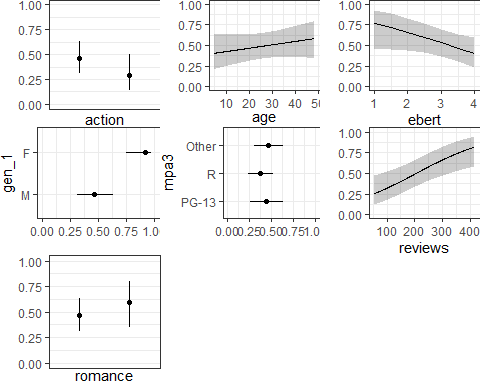
## Plotting the Effects for fit2\_lrm

plot(summary(fit2\_lrm))



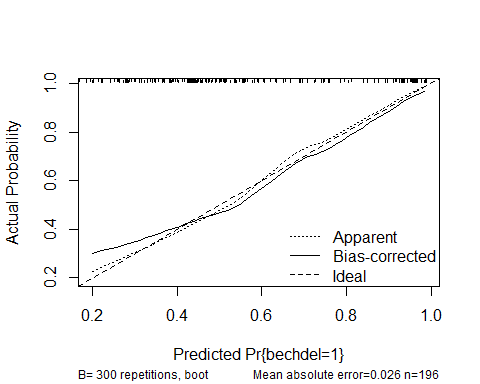
## Prediction Plots for fit2\_lrm

ggplot(Predict(fit2\_lrm, fun = plogis))



## Calibration Plot for fit2\_lrm

set.seed(4321232)  
plot(calibrate(fit2\_lrm, method = "boot", B = 300))



n=196 Mean absolute error=0.026 Mean squared error=0.00112  
0.9 Quantile of absolute error=0.053

## fit2 Hosmer-Lemeshow test

* Here we’ll revert to the default choice of 10 bins.

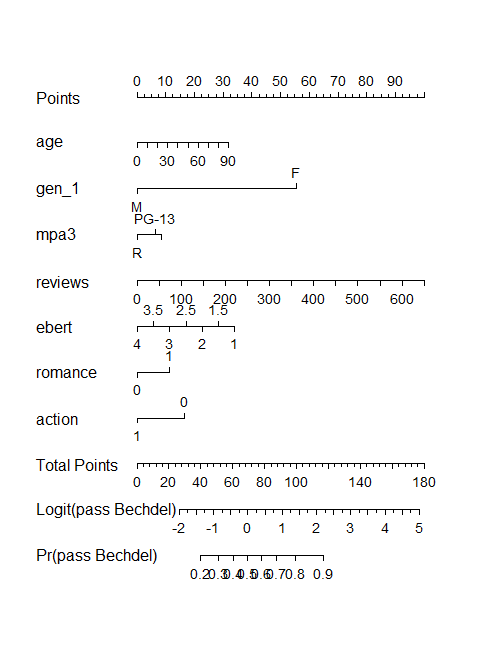
performance\_hosmer(fit2, n\_bins = 10)

# Hosmer-Lemeshow Goodness-of-Fit Test  
  
 Chi-squared: 8.822  
 df: 8   
 p-value: 0.358

Summary: model seems to fit well.

## Nomogram for fit2\_lrm

plot(nomogram(fit2\_lrm, fun = plogis, funlabel = "Pr(pass Bechdel)"),  
 lplabel = "Logit(pass Bechdel)")



## Predictions from fit2 for love4 movies

augment(fit2, newdata = love4, type.predict = "response") |>  
 select(movie, .fitted, bechdel, everything()) |>  
 gt() |> tab\_options(table.font.size = 20) |>  
 opt\_stylize(style = 5, color = "pink")

| movie | .fitted | bechdel | mov\_id | age | gen\_1 | mpa3 | reviews | ebert | romance | action |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The Manchurian Candidate | 0.3438133 | 0 | L-2 | 63 | M | PG-13 | 67 | 4.0 | 0 | 0 |
| Die Hard | 0.2839742 | 0 | L-8 | 37 | M | R | 89 | 2.0 | 0 | 1 |
| Love Actually | 0.6751826 | 1 | L-63 | 22 | M | Other | 230 | 3.5 | 1 | 0 |
| Sense and Sensibility | 0.9075621 | 1 | L-125 | 30 | F | Other | 67 | 3.5 | 1 | 0 |

## CIs around our predictions?

augment(fit2, newdata = love4, type.predict = "link", se\_fit = TRUE) |>  
 mutate(ci\_90\_low = .fitted - 1.645 \* .se.fit,   
 ci\_90\_high = .fitted + 1.645 \* .se.fit) |>  
 select(movie, .fitted, .se.fit, ci\_90\_low, ci\_90\_high, bechdel, everything())

# A tibble: 4 × 14  
 movie .fitted .se.fit ci\_90\_low ci\_90\_high bechdel mov\_id age gen\_1 mpa3   
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <fct> <fct>  
1 The Man… -0.646 0.719 -1.83 0.536 0 L-2 63 M PG-13  
2 Die Hard -0.925 0.620 -1.95 0.0957 0 L-8 37 M R   
3 Love Ac… 0.732 0.517 -0.118 1.58 1 L-63 22 M Other  
4 Sense a… 2.28 0.732 1.08 3.49 1 L-125 30 F Other  
# ℹ 4 more variables: reviews <dbl>, ebert <dbl>, romance <dbl>, action <dbl>

## Converting from Logit to Probability Scale

For Sense and Sensibility, our predicted logit(bechdel pass) = 2.2842, with 90% CI (1.0809, 3.4876).

* If logit(bechdel pass) = 2.2842, then odds(bechdel pass) = exp(2.2842), and pr(bechdel pass) = exp(2.2842) / (1 + exp(2.2842)) = 9.8178 / 10.8178 = 0.908
* If logit(bechdel pass) = 1.0809, then odds(bechdel pass) = exp(1.0809), and pr(bechdel pass) = exp(1.0809) / (1 + exp(1.0809)) = 2.9473 / 3.9473 = 0.747
* If logit(bechdel pass) = 3.4876, then odds(bechdel pass) = exp(3.4876), and pr(bechdel pass) = exp(3.4876) / (1 + exp(3.4876)) = 32.7074 / 33.7074 = 0.970

Predicted prob(bechdel pass) = 0.908 with 90% confidence interval (0.747, 0.970) for Sense and Sensibility using fit2.

## Picking a Decision Rule for fit2

* Again, using cutpointr to select a decision rule which maximizes “Sensitivity” + “Specificity”.

fit2\_aug <- augment(fit2, type.predict = "response")  
  
cp2 <- cutpointr(data = fit2\_aug, .fitted, bechdel,   
 pos\_class = 1, neg\_class = 0,  
 method = maximize\_metric, metric = sum\_sens\_spec)

Assuming the positive class has higher x values

cp2 |> select(direction, optimal\_cutpoint, method, sum\_sens\_spec) |>   
 gt() |> tab\_options(table.font.size = 24) |>   
 opt\_stylize(style = 2, color = "pink")

| direction | optimal\_cutpoint | method | sum\_sens\_spec |
| --- | --- | --- | --- |
| >= | 0.6950129 | maximize\_metric | 1.494829 |

## Confusion Matrix for fit2

cm2 <- confusionMatrix(data = factor(fit2\_aug$.fitted >= cp2$optimal\_cutpoint),  
 reference = factor(fit2\_aug$bechdel == 1), positive = "TRUE")  
cm2

Confusion Matrix and Statistics  
  
 Reference  
Prediction FALSE TRUE  
 FALSE 80 53  
 TRUE 3 60  
   
 Accuracy : 0.7143   
 95% CI : (0.6456, 0.7764)  
 No Information Rate : 0.5765   
 P-Value [Acc > NIR] : 4.645e-05   
   
 Kappa : 0.4582   
   
 Mcnemar's Test P-Value : 5.835e-11   
   
 Sensitivity : 0.5310   
 Specificity : 0.9639   
 Pos Pred Value : 0.9524   
 Neg Pred Value : 0.6015   
 Prevalence : 0.5765   
 Detection Rate : 0.3061   
 Detection Prevalence : 0.3214   
 Balanced Accuracy : 0.7474   
   
 'Positive' Class : TRUE

## fit2 PCP

Percentage of Correct Predictions (with 0.5 decision rule)

performance\_pcp(fit2, ci = 0.90, method = "Herron")

# Percentage of Correct Predictions from Logistic Regression Model  
  
 Full model: 63.62% [57.96% - 69.27%]  
 Null model: 51.17% [45.30% - 57.04%]  
  
# Likelihood-Ratio-Test  
  
 Chi-squared: 57.119  
 df: 8.000  
 p-value: 0.000

# Checking Assumptions in Logistic Regression Models

## Linear Regression vs. Logistic Regression

Adapted from <https://www.statology.org/assumptions-of-logistic-regression/>

In contrast to linear regression, logistic regression does not require:

* A linear relationship between the predictors and the outcome.
* The residuals of the model to be normally distributed.
* The residuals to have constant variance (homoscedasticity/)

## Assumptions of Logistic Regression

Adapted from <https://www.statology.org/assumptions-of-logistic-regression/>

1. The outcome variable is binary.
2. The observations are independent from each other. (They shouldn’t show a pattern in time or space.)
3. There is no severe multicollinearity among the predictors (we use VIF > 5 as an indicator of trouble.)

## Assumptions of Logistic Regression

1. There are no extreme outliers (Cook’s distance > .5 is what R flags as problematic[[2]](#footnote-158).)
2. The sample size is sufficiently large (see next few slides.)
3. There is a linear relationship between predictors and the logit of the outcome (see the final few slides.)

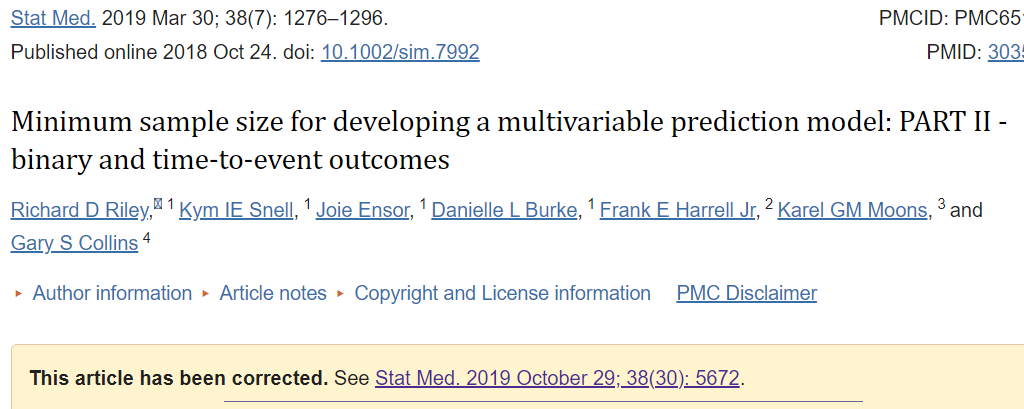
## What does sufficiently large mean?

1. Some people like a simple rule like 500 observations overall and 10 events (where an event is the smaller of your two outcome groups) per predictor parameter. See [Long’s 1997 book (pdf)](https://jslsoc.sitehost.iu.edu/files_research/rm4cldv/sage1997/rm4cldv_toc.pdf).

For *Project A*, we focus on keeping the number of predictors below (4 + (N-100)) / 100) where N is the size of the smaller of your two outcome groups. I wouldn’t use that standard outside of Project A, though.

## What does sufficiently large mean?

1. Riley et al. in [Statistics in Medicine](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6519266/) develop an estimation scheme for the needed sample sizes, and motivate it with several examples. It’s pretty complex but it’s a good option.



## First Two Assumptions

1. The outcome variable is binary.
   * OK. bechdel is either 1 (Pass) or 0 (Fail.)

mov25 |> count(bechdel)

# A tibble: 2 × 2  
 bechdel n  
 <fct> <int>  
1 0 83  
2 1 113

1. The observations are independent from each other. (They shouldn’t show a pattern in time or space.)
   * The data are cross-sectional. No one film’s results should affect another film’s results, so we’re OK.

## Assumption Three

1. There is no severe multicollinearity among the predictors (we use VIF > 5 as an indicator of trouble.)

car::vif(fit2)

GVIF Df GVIF^(1/(2\*Df))  
age 2.058811 1 1.434856  
gen\_1 1.031006 1 1.015385  
mpa3 1.390834 2 1.085973  
reviews 2.345235 1 1.531416  
ebert 1.279741 1 1.131257  
romance 1.131410 1 1.063678  
action 1.352148 1 1.162819

rms::vif(fit2\_lrm)

age gen\_1=F mpa3=R mpa3=Other reviews ebert romance   
 2.058813 1.031005 1.635040 1.657426 2.345235 1.279742 1.131409   
 action   
 1.352148

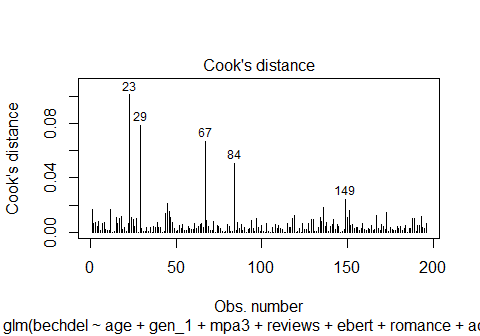
## Assumption Four

1. There are no extreme outliers (no Cook’s distance > 0.5)

max(cooks.distance(fit2))

[1] 0.1011299

plot(fit2, which = 4, id.n = 5)



## Assumption Five

1. The sample size is sufficiently large.

* Recall that we have 113 Pass and 83 Fail movies in mov25.

glance(fit2) |> select(nobs) ## could use mod2\_lrm$stats["Obs"]

# A tibble: 1 × 1  
 nobs  
 <int>  
1 196

Does this seem like enough observations to fit a logistic regression model with 7 predictors (and 8 df) under consideration?

## Assumption Six

1. There is a linear relationship between predictors and the logit of the outcome.

A **Box-Tidwell test** is a common strategy to test this assumption, but it doesn’t work for logistic models [according to John Fox](https://stackoverflow.com/questions/56350546/how-to-use-the-box-tidwell-function-with-a-logistic-regression-in-r), inventor of the car package[[3]](#footnote-175).

He instead recommends Component + Residual plots (Partial Residual plots) which can be used for linear and generalized linear models.

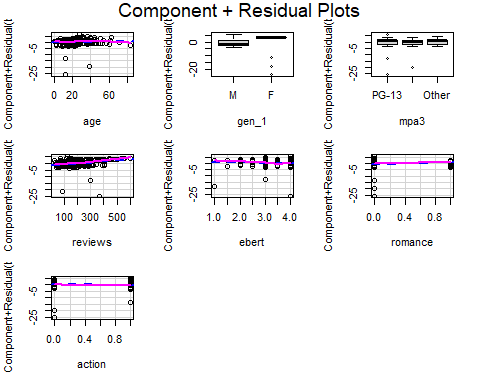
## Interpreting the Partial Residual Plots

* The blue dashed line shows the expected residuals if the relationship between the predictor and response variable (here the log odds of our outcome) was linear.
* The solid pink curve shows a loess smooth of the actual residuals.

If the two lines are meaningfully different, then this is evidence of a nonlinear relationship. One way to fix this issue is to build a transformation on the predictor variables, or consider incorporating some non-linear terms.

## Running Partial Residual Plots

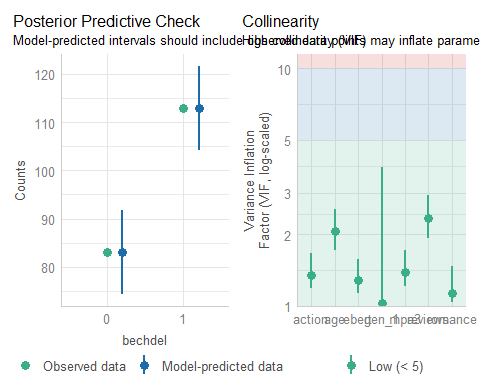
crPlots(fit2) ## crPlots comes from the car package



## check\_model for fit2 (1/4)

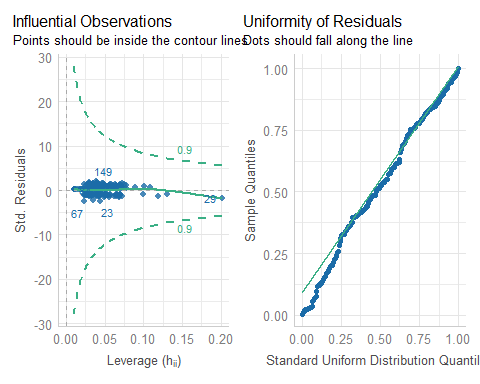
check\_model(fit2, check = c("pp\_check", "vif"))

Cannot simulate residuals for models of class `glm`. Please try  
 `check\_model(..., residual\_type = "normal")` instead.



## check\_model for fit2 (2/4)

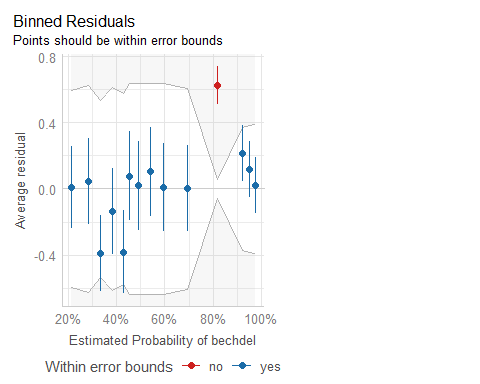
check\_model(fit2, check = c("outliers", "qq"))



## check\_model for fit2 (3/4)

check\_model(fit2, check = c("binned\_residuals"))

Cannot simulate residuals for models of class `glm`. Please try  
 `check\_model(..., residual\_type = "normal")` instead.



## check\_model for fit2 (4/4)

* Extra details for the last three plots…

check\_outliers(fit2)

OK: No outliers detected.  
- Based on the following method and threshold: cook (0.9).  
- For variable: (Whole model)

check\_residuals(fit2)

OK: Simulated residuals appear as uniformly distributed (p = 0.234).

binned\_residuals(fit2)

Warning: About 93% of the residuals are inside the error bounds (~95% or higher would be good).

## Analysis of Deviance for fit2

anova(fit2\_lrm)

Wald Statistics Response: bechdel   
  
 Factor Chi-Square d.f. P   
 age 0.94 1 0.3316  
 gen\_1 16.31 1 0.0001  
 mpa3 0.93 2 0.6271  
 reviews 9.37 1 0.0022  
 ebert 3.75 1 0.0527  
 romance 1.24 1 0.2653  
 action 3.19 1 0.0741  
 TOTAL 32.63 8 0.0001

|  |
| --- |
| Note |
| Remember that this result shows sequential tests, and if you change the order of the predictors, the *p* values will change. |

## Validating Key Summaries (fit2)

set.seed(202502062); validate(fit2\_lrm, B = 300)

index.orig training test optimism index.corrected n  
Dxy 0.5842 0.6152 0.5471 0.0681 0.5161 300  
R2 0.3398 0.3780 0.3056 0.0724 0.2674 300  
Intercept 0.0000 0.0000 -0.0152 0.0152 -0.0152 300  
Slope 1.0000 1.0000 0.7964 0.2036 0.7964 300  
Emax 0.0000 0.0000 0.0533 0.0533 0.0533 300  
D 0.2863 0.3265 0.2531 0.0734 0.2129 300  
U -0.0102 -0.0102 0.0283 -0.0385 0.0283 300  
Q 0.2965 0.3367 0.2247 0.1119 0.1846 300  
B 0.1811 0.1725 0.1903 -0.0178 0.1989 300  
g 1.6424 2.0734 1.4809 0.5925 1.0499 300  
gp 0.2827 0.2987 0.2668 0.0318 0.2509 300

* C = 0.5 + Dxy, so validated C for fit2 = 0.5 + (0.5161/2) = 0.7581, validated Nagelkerke = 0.2674, and validated Brier score B = 0.1989

## Cross-Validating AUC for fit2

set.seed(123452)  
performance\_accuracy(fit2, method = "cv", k = 5, ci = 0.90)

# Accuracy of Model Predictions  
  
Accuracy (90% CI): 75.94% [69.47%, 84.70%]  
Method: Area under Curve

# Model fit3

## Preparing to Search through our predictors

mov25\_sf <-   
 cobalt::splitfactor(mov25, "mpa3", replace = TRUE, drop.first = TRUE)  
  
names(mov25\_sf)

[1] "mov\_id" "bechdel" "age" "mpa3\_R" "mpa3\_Other"  
 [6] "reviews" "ebert" "gen\_1" "romance" "action"   
[11] "movie"

Xy <- mov25\_sf |>   
 select(age, mpa3\_R, mpa3\_Other, reviews, ebert,   
 gen\_1, romance, action, bechdel) |>  
 data.frame()

* We’re building Xy to contain all of our predictors (with multi-categorical predictors inserted using indicator variables) followed by the outcome.

## Predictor Subset with the best AIC

The code below searches the predictors in Xy for the combination that produces the smallest (hence best) AIC (Akaike Information Criterion).

best\_AIC <- bestglm(Xy, IC = "AIC", family = binomial)

Morgan-Tatar search since family is non-gaussian.

best\_AIC

AIC  
BICq equivalent for q in (0.726461340395408, 0.861227305047834)  
Best Model:  
 Estimate Std. Error z value Pr(>|z|)  
(Intercept) 0.529989558 0.827647052 0.640357 5.219406e-01  
reviews 0.005135746 0.001600168 3.209506 1.329635e-03  
ebert -0.465685210 0.250830225 -1.856575 6.337160e-02  
gen\_1F 2.711777250 0.632437155 4.287821 1.804345e-05  
action -0.715810359 0.381228674 -1.877640 6.043038e-02

## Model fit3: best subsets with AIC

fit3 <- glm(bechdel ~ gen\_1 + reviews + ebert + action,  
 data = mov25, family = binomial(link = "logit"))  
  
n\_obs(fit3)

[1] 196

performance\_roc(fit3)

AUC: 77.92%

model\_performance(fit3)

# Indices of model performance  
  
AIC | AICc | BIC | Tjur's R2 | RMSE | Sigma | Log\_loss | Score\_log  
------------------------------------------------------------------------------  
223.684 | 224.000 | 240.075 | 0.240 | 0.429 | 1.000 | 0.545 | -132.860  
  
AIC | Score\_spherical | PCP  
---------------------------------  
223.684 | 0.008 | 0.629

## Model fit3 parameters

model\_parameters(fit3, exponentiate = TRUE, ci = 0.90)

Parameter | Odds Ratio | SE | 90% CI | z | p  
--------------------------------------------------------------------  
(Intercept) | 1.70 | 1.41 | [0.44, 6.79] | 0.64 | 0.522   
gen 1 [F] | 15.06 | 9.52 | [5.91, 49.67] | 4.29 | < .001  
reviews | 1.01 | 1.61e-03 | [1.00, 1.01] | 3.21 | 0.001   
ebert | 0.63 | 0.16 | [0.41, 0.94] | -1.86 | 0.063   
action | 0.49 | 0.19 | [0.26, 0.91] | -1.88 | 0.060

Uncertainty intervals (profile-likelihood) and p-values (two-tailed)  
 computed using a Wald z-distribution approximation.

## What is the fit3 equation?

fit3$coefficients ## note: without exponentiation

(Intercept) gen\_1F reviews ebert action   
 0.529989558 2.711777250 0.005135746 -0.465685210 -0.715810359

$$
logit(\mbox{bechdel = 1}) = \\
0.5300 + 2.7118 (\mbox{gen\_1 = F}) + 0.0051 (\mbox{reviews}) - \\
0.4657 (\mbox{ebert}) - 0.7158 (\mbox{action})
$$

## lrm version of fit3

d <- datadist(mov25); options(datadist = "d")  
  
fit3\_lrm <- lrm(bechdel ~ gen\_1 + reviews + ebert + action,   
 data = mov25, x = TRUE, y = TRUE)

|  |
| --- |
| Key Summaries for fit3\_lrm include… |
| * C = 0.779, Nagelkerke = 0.321, Brier score = 0.184 * See next slide for details. |

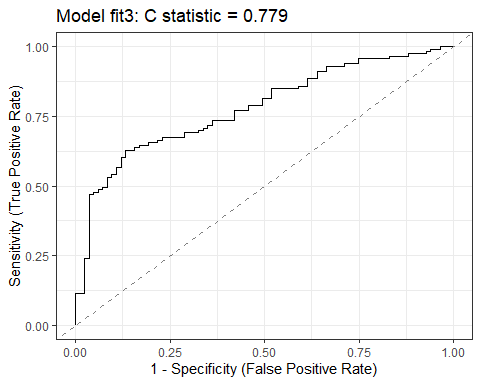
## fit3\_lrm summaries

fit3\_lrm

Logistic Regression Model  
  
lrm(formula = bechdel ~ gen\_1 + reviews + ebert + action, data = mov25,   
 x = TRUE, y = TRUE)  
  
 Model Likelihood Discrimination Rank Discrim.   
 Ratio Test Indexes Indexes   
Obs 196 LR chi2 53.42 R2 0.321 C 0.779   
 0 83 d.f. 4 R2(4,196)0.223 Dxy 0.558   
 1 113 Pr(> chi2) <0.0001 R2(4,143.6)0.291 gamma 0.558   
max |deriv| 5e-05 Brier 0.184 tau-a 0.274   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept 0.5300 0.8276 0.64 0.5219   
gen\_1=F 2.7118 0.6324 4.29 <0.0001   
reviews 0.0051 0.0016 3.21 0.0013   
ebert -0.4657 0.2508 -1.86 0.0634   
action -0.7158 0.3812 -1.88 0.0604

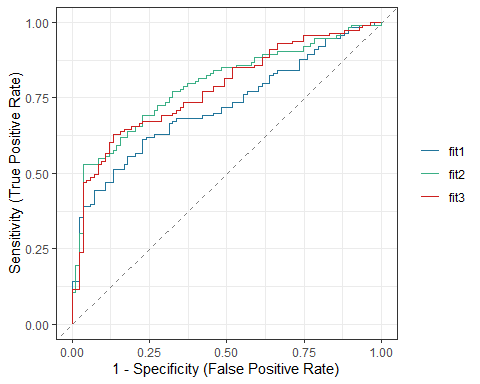
## ROC Curve Analysis for fit3

plot(performance\_roc(fit3)) +  
 labs(title = glue("Model fit3: C statistic = ",  
 round\_half\_up(as.numeric(performance\_roc(fit3)),3)))



## ROC curves for models so far

plot(performance\_roc(fit1, fit2, fit3)) +  
 scale\_color\_social()



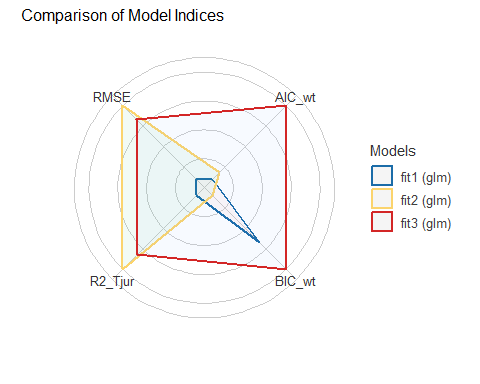
## Pseudo- measures (models so far)

r2\_res123 <- tibble(name = c("fit1", "fit2", "fit3"),   
 McFadden = c(as.numeric(r2\_mcfadden(fit1)[1]),  
 as.numeric(r2\_mcfadden(fit2)[1]),  
 as.numeric(r2\_mcfadden(fit3)[1])),  
 Nagelkerke = c(as.numeric(r2\_nagelkerke(fit1)),  
 as.numeric(r2\_nagelkerke(fit2)),  
 as.numeric(r2\_nagelkerke(fit3))),  
 Tjur = c(as.numeric(r2\_tjur(fit1)),  
 as.numeric(r2\_tjur(fit2)),  
 as.numeric(r2\_tjur(fit3))))  
  
r2\_res123 |> gt() |> fmt\_number(decimals = 4) |>   
 tab\_options(table.font.size = 24) |>  
 opt\_stylize(style = 4, color = "green")

| name | McFadden | Nagelkerke | Tjur |
| --- | --- | --- | --- |
| fit1 | 0.1571 | 0.2590 | 0.1813 |
| fit2 | 0.2138 | 0.3398 | 0.2549 |
| fit3 | 0.2000 | 0.3206 | 0.2397 |

## Comparing Model Indices from our 3 models

plot(compare\_performance(fit1, fit2, fit3, metrics = "common"))



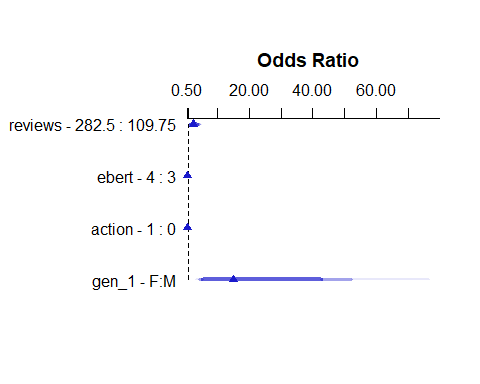
## What are the effect sizes in fit3\_lrm?

summary(fit3\_lrm)

Effects Response : bechdel   
  
 Factor Low High Diff. Effect S.E. Lower 0.95 Upper 0.95  
 reviews 109.75 282.5 172.75 0.88720 0.27643 0.34541 1.429000   
 Odds Ratio 109.75 282.5 172.75 2.42830 NA 1.41260 4.174500   
 ebert 3.00 4.0 1.00 -0.46569 0.25083 -0.95730 0.025934   
 Odds Ratio 3.00 4.0 1.00 0.62770 NA 0.38393 1.026300   
 action 0.00 1.0 1.00 -0.71581 0.38123 -1.46300 0.031385   
 Odds Ratio 0.00 1.0 1.00 0.48880 NA 0.23154 1.031900   
 gen\_1 - F:M 1.00 2.0 NA 2.71180 0.63245 1.47220 3.951400   
 Odds Ratio 1.00 2.0 NA 15.05600 NA 4.35880 52.006000

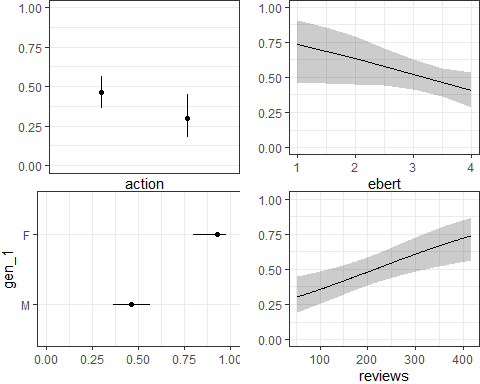
## Plotting the Effects for fit3\_lrm

plot(summary(fit3\_lrm))



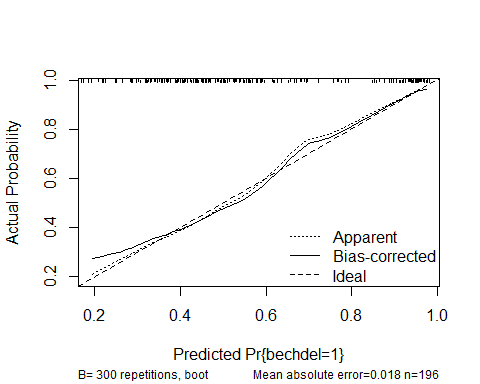
## Prediction Plots for fit3\_lrm

ggplot(Predict(fit3\_lrm, fun = plogis))



## Calibration Plot for fit3\_lrm

set.seed(4321233)  
plot(calibrate(fit3\_lrm, method = "boot", B = 300))



n=196 Mean absolute error=0.018 Mean squared error=0.00052  
0.9 Quantile of absolute error=0.033

## fit3 Hosmer-Lemeshow test

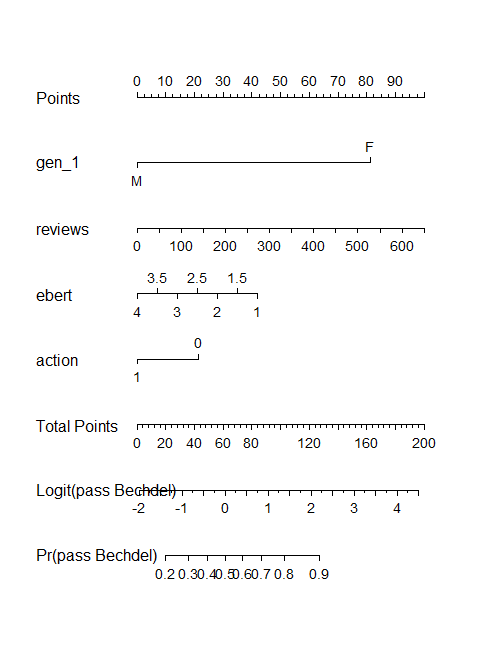
performance\_hosmer(fit3, n\_bins = 10)

# Hosmer-Lemeshow Goodness-of-Fit Test  
  
 Chi-squared: 10.006  
 df: 8   
 p-value: 0.265

Summary: model seems to fit well.

## Nomogram for fit3\_lrm

plot(nomogram(fit3\_lrm, fun = plogis, funlabel = "Pr(pass Bechdel)"),  
 lplabel = "Logit(pass Bechdel)")



## Predictions from fit3 for love4 movies

augment(fit3, newdata = love4, type.predict = "response") |>  
 select(movie, .fitted, bechdel, everything()) |>  
 gt() |> tab\_options(table.font.size = 20) |>  
 opt\_stylize(style = 5, color = "pink")

| movie | .fitted | bechdel | mov\_id | age | gen\_1 | mpa3 | reviews | ebert | romance | action |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The Manchurian Candidate | 0.2711776 | 0 | L-2 | 63 | M | PG-13 | 67 | 4.0 | 0 | 0 |
| Die Hard | 0.3407150 | 0 | L-8 | 37 | M | R | 89 | 2.0 | 0 | 1 |
| Love Actually | 0.5203171 | 1 | L-63 | 22 | M | Other | 230 | 3.5 | 1 | 0 |
| Sense and Sensibility | 0.8760955 | 1 | L-125 | 30 | F | Other | 67 | 3.5 | 1 | 0 |

## CIs around our predictions?

augment(fit3, newdata = love4, type.predict = "link", se\_fit = TRUE) |>  
 mutate(ci\_90\_low = .fitted - 1.645 \* .se.fit,   
 ci\_90\_high = .fitted + 1.645 \* .se.fit) |>  
 select(movie, .fitted, .se.fit, ci\_90\_low, ci\_90\_high, bechdel, everything())

# A tibble: 4 × 14  
 movie .fitted .se.fit ci\_90\_low ci\_90\_high bechdel mov\_id age gen\_1 mpa3   
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <fct> <fct>  
1 The Man… -0.989 0.358 -1.58 -0.399 0 L-2 63 M PG-13  
2 Die Hard -0.660 0.461 -1.42 0.0980 0 L-8 37 M R   
3 Love Ac… 0.0813 0.215 -0.272 0.435 1 L-63 22 M Other  
4 Sense a… 1.96 0.636 0.909 3.00 1 L-125 30 F Other  
# ℹ 4 more variables: reviews <dbl>, ebert <dbl>, romance <dbl>, action <dbl>

## Converting from Logit to Probability Scale

For Love Actually, our predicted logit(bechdel pass) = 0.0813, with 90% CI (-0.2725, 0.4351).

* If logit(bechdel pass) = 0.0813, then odds(bechdel pass) = exp(0.0813), and pr(bechdel pass) = exp(0.0813) / (1 + exp(0.0813)) = 1.0847/2.0847 = 0.520
* If logit(bechdel pass) = -0.2725, then odds(bechdel pass) = exp(-0.2725), and pr(bechdel pass) = exp(-0.2725) / (1 + exp(-0.2725)) = 0.7615/1.7615 = 0.432
* If logit(bechdel pass) = 0.4351, then odds(bechdel pass) = exp(0.4351), and pr(bechdel pass) = exp(0.4351) / (1 + exp(0.4351)) = 1.5451 / 2.5451 = 0.607

Predicted prob(bechdel pass) = 0.520 with 90% confidence interval (0.432, 0.607) for Love Actually using fit3.

## Picking a Decision Rule for fit3

* Again, using cutpointr to select a decision rule which maximizes “Sensitivity” + “Specificity”.

fit3\_aug <- augment(fit3, type.predict = "response")  
  
cp3 <- cutpointr(data = fit3\_aug, .fitted, bechdel,   
 pos\_class = 1, neg\_class = 0,  
 method = maximize\_metric, metric = sum\_sens\_spec)

Assuming the positive class has higher x values

cp3 |> select(direction, optimal\_cutpoint, method, sum\_sens\_spec) |>   
 gt() |> tab\_options(table.font.size = 24) |>   
 opt\_stylize(style = 2, color = "pink")

| direction | optimal\_cutpoint | method | sum\_sens\_spec |
| --- | --- | --- | --- |
| >= | 0.5631739 | maximize\_metric | 1.495788 |

## Confusion Matrix for fit3

cm3 <- confusionMatrix(data = factor(fit3\_aug$.fitted >= cp3$optimal\_cutpoint),  
 reference = factor(fit3\_aug$bechdel == 1), positive = "TRUE")  
cm3

Confusion Matrix and Statistics  
  
 Reference  
Prediction FALSE TRUE  
 FALSE 72 42  
 TRUE 11 71  
   
 Accuracy : 0.7296   
 95% CI : (0.6617, 0.7904)  
 No Information Rate : 0.5765   
 P-Value [Acc > NIR] : 6.311e-06   
   
 Kappa : 0.4724   
   
 Mcnemar's Test P-Value : 3.775e-05   
   
 Sensitivity : 0.6283   
 Specificity : 0.8675   
 Pos Pred Value : 0.8659   
 Neg Pred Value : 0.6316   
 Prevalence : 0.5765   
 Detection Rate : 0.3622   
 Detection Prevalence : 0.4184   
 Balanced Accuracy : 0.7479   
   
 'Positive' Class : TRUE

## fit3 PCP

Percentage of Correct Predictions (with 0.5 decision rule)

performance\_pcp(fit3, ci = 0.90, method = "Herron")

# Percentage of Correct Predictions from Logistic Regression Model  
  
 Full model: 62.88% [57.20% - 68.55%]  
 Null model: 51.17% [45.30% - 57.04%]  
  
# Likelihood-Ratio-Test  
  
 Chi-squared: 53.419  
 df: 4.000  
 p-value: 0.000

## No severe multicollinearity? (fit3)

car::vif(fit3)

gen\_1 reviews ebert action   
1.017028 1.160355 1.089948 1.088776

rms::vif(fit3\_lrm)

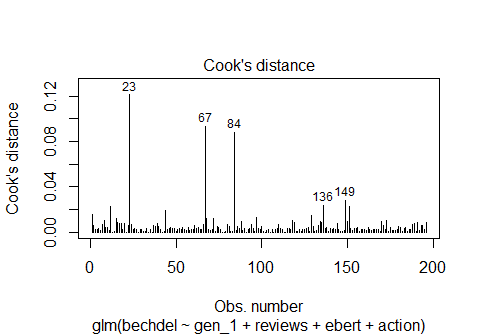
gen\_1=F reviews ebert action   
1.017028 1.160355 1.089947 1.088776

## No Cook’s distance > 0.5? (fit3)

max(cooks.distance(fit3))

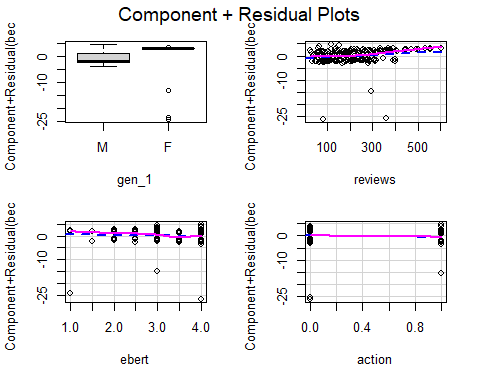
[1] 0.121117

plot(fit3, which = 4, id.n = 5)



## fit3 Partial Residual Plots

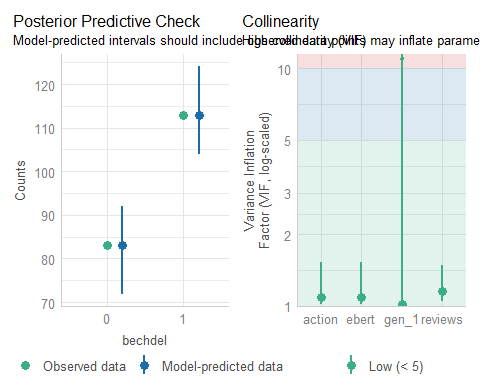
crPlots(fit3) ## crPlots comes from the car package



## check\_model for fit3 (1/4)

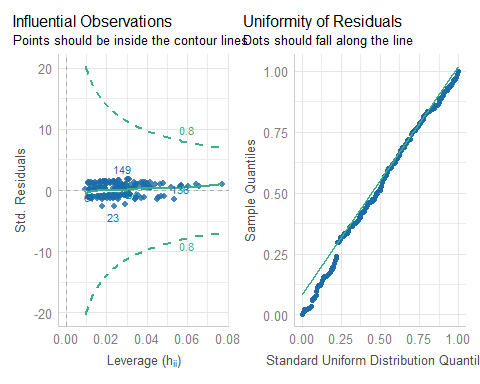
check\_model(fit3, check = c("pp\_check", "vif"))

Cannot simulate residuals for models of class `glm`. Please try  
 `check\_model(..., residual\_type = "normal")` instead.



## check\_model for fit3 (2/4)

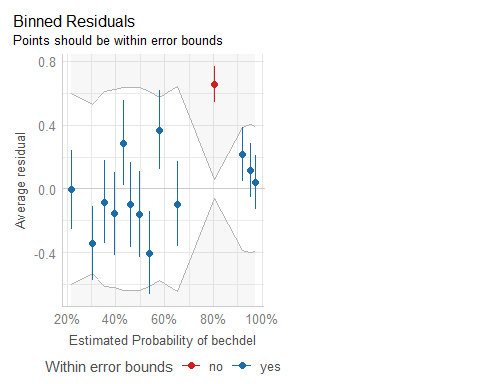
check\_model(fit3, check = c("outliers", "qq"))



## check\_model for fit3 (3/4)

check\_model(fit3, check = c("binned\_residuals"))

Cannot simulate residuals for models of class `glm`. Please try  
 `check\_model(..., residual\_type = "normal")` instead.



## check\_model for fit3 (4/4)

* Extra details for the last three plots…

check\_outliers(fit3)

OK: No outliers detected.  
- Based on the following method and threshold: cook (0.8).  
- For variable: (Whole model)

check\_residuals(fit3)

OK: Simulated residuals appear as uniformly distributed (p = 0.265).

binned\_residuals(fit3)

Warning: About 93% of the residuals are inside the error bounds (~95% or higher would be good).

## Analysis of Deviance for fit3

anova(fit3\_lrm)

Wald Statistics Response: bechdel   
  
 Factor Chi-Square d.f. P   
 gen\_1 18.38 1 <.0001  
 reviews 10.30 1 0.0013  
 ebert 3.45 1 0.0634  
 action 3.53 1 0.0604  
 TOTAL 30.70 4 <.0001

|  |
| --- |
| Note |
| Remember that this result shows sequential tests, and if you change the order of the predictors, the *p* values will change. |

## Validating Key Summaries (fit3)

set.seed(202502063); validate(fit3\_lrm, B = 300)

index.orig training test optimism index.corrected n  
Dxy 0.5584 0.5668 0.5423 0.0245 0.5339 300  
R2 0.3206 0.3349 0.3047 0.0301 0.2905 300  
Intercept 0.0000 0.0000 0.0038 -0.0038 0.0038 300  
Slope 1.0000 1.0000 0.9194 0.0806 0.9194 300  
Emax 0.0000 0.0000 0.0194 0.0194 0.0194 300  
D 0.2674 0.2831 0.2521 0.0309 0.2365 300  
U -0.0102 -0.0102 0.0182 -0.0284 0.0182 300  
Q 0.2776 0.2933 0.2339 0.0593 0.2183 300  
B 0.1845 0.1811 0.1893 -0.0082 0.1927 300  
g 1.5573 1.8436 1.4784 0.3652 1.1921 300  
gp 0.2715 0.2757 0.2629 0.0128 0.2587 300

* C = 0.5 + Dxy, so validated C for fit3 = 0.5 + (0.5339/2) = 0.7670, validated Nagelkerke = 0.2905, and validated Brier score B = 0.1927

## Cross-Validating AUC for fit3

set.seed(123453)  
performance\_accuracy(fit3, method = "cv", k = 5, ci = 0.90)

# Accuracy of Model Predictions  
  
Accuracy (90% CI): 74.33% [66.96%, 84.85%]  
Method: Area under Curve

# Model fit4

## Preparing to Search through our predictors

mov25\_sf <-   
 cobalt::splitfactor(mov25, "mpa3", replace = TRUE, drop.first = TRUE)  
  
names(mov25\_sf)

[1] "mov\_id" "bechdel" "age" "mpa3\_R" "mpa3\_Other"  
 [6] "reviews" "ebert" "gen\_1" "romance" "action"   
[11] "movie"

Xy <- mov25\_sf |>   
 select(age, mpa3\_R, mpa3\_Other, reviews, ebert,   
 gen\_1, romance, action, bechdel) |>  
 data.frame()

Just as before, Xy contains all predictors (with multi-categorical predictors inserted using indicator variables) followed by the outcome.

## Predictor Subset with the best BIC

The code below searches the predictors in Xy for the combination that produces the smallest (hence best) BIC (Bayes Information Criterion).

best\_BIC <- bestglm(Xy, IC = "BIC", family = binomial)

Morgan-Tatar search since family is non-gaussian.

best\_BIC

BIC  
BICq equivalent for q in (0.227337090988041, 0.726461340395408)  
Best Model:  
 Estimate Std. Error z value Pr(>|z|)  
(Intercept) -0.959364552 0.345824391 -2.774138 5.534821e-03  
reviews 0.003885784 0.001452304 2.675600 7.459551e-03  
gen\_1F 2.926645287 0.626750302 4.669555 3.018522e-06

## Model fit4: best subsets with BIC

fit4 <- glm(bechdel ~ gen\_1 + reviews,  
 data = mov25, family = binomial(link = "logit"))  
  
n\_obs(fit4)

[1] 196

performance\_roc(fit4)

AUC: 74.30%

model\_performance(fit4)

# Indices of model performance  
  
AIC | AICc | BIC | Tjur's R2 | RMSE | Sigma | Log\_loss | Score\_log  
------------------------------------------------------------------------------  
226.334 | 226.459 | 236.168 | 0.208 | 0.439 | 1.000 | 0.562 | -129.434  
  
AIC | Score\_spherical | PCP  
---------------------------------  
226.334 | 0.007 | 0.613

## Model fit4 parameters

model\_parameters(fit4, exponentiate = TRUE, ci = 0.90)

Parameter | Odds Ratio | SE | 90% CI | z | p  
--------------------------------------------------------------------  
(Intercept) | 0.38 | 0.13 | [0.21, 0.67] | -2.77 | 0.006   
gen 1 [F] | 18.66 | 11.70 | [7.41, 61.16] | 4.67 | < .001  
reviews | 1.00 | 1.46e-03 | [1.00, 1.01] | 2.68 | 0.007

Uncertainty intervals (profile-likelihood) and p-values (two-tailed)  
 computed using a Wald z-distribution approximation.

## What is the fit4 equation?

fit4$coefficients ## note: without exponentiation

(Intercept) gen\_1F reviews   
-0.959364552 2.926645287 0.003885784

$$
logit(\mbox{bechdel = 1}) = \\
-0.9594 + 2.9266 (\mbox{gen\_1 = F}) + 0.0039 (\mbox{reviews})
$$

## lrm version of fit4

d <- datadist(mov25); options(datadist = "d")  
  
fit4\_lrm <- lrm(bechdel ~ gen\_1 + reviews,   
 data = mov25, x = TRUE, y = TRUE)

|  |
| --- |
| Key Summaries for fit4\_lrm include… |
| * C = 0.743, Nagelkerke = 0.285, Brier score = 0.192 * See next slide for details. |

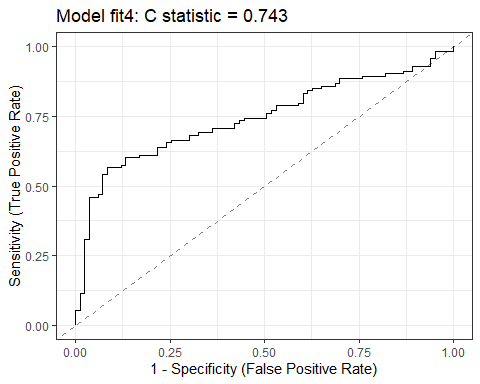
## fit4\_lrm summaries

fit4\_lrm

Logistic Regression Model  
  
lrm(formula = bechdel ~ gen\_1 + reviews, data = mov25, x = TRUE,   
 y = TRUE)  
  
 Model Likelihood Discrimination Rank Discrim.   
 Ratio Test Indexes Indexes   
Obs 196 LR chi2 46.77 R2 0.285 C 0.743   
 0 83 d.f. 2 R2(2,196)0.204 Dxy 0.486   
 1 113 Pr(> chi2) <0.0001 R2(2,143.6)0.268 gamma 0.487   
max |deriv| 3e-05 Brier 0.192 tau-a 0.239   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept -0.9594 0.3458 -2.77 0.0055   
gen\_1=F 2.9266 0.6268 4.67 <0.0001   
reviews 0.0039 0.0015 2.68 0.0075

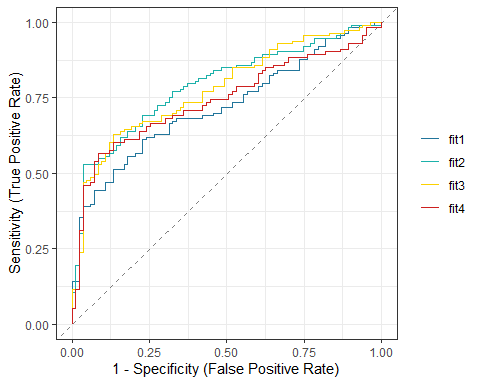
## ROC Curve Analysis for fit4

plot(performance\_roc(fit4)) +  
 labs(title = glue("Model fit4: C statistic = ",  
 round\_half\_up(as.numeric(performance\_roc(fit4)),3)))



## ROC curves for models so far

plot(performance\_roc(fit1, fit2, fit3, fit4)) +  
 scale\_color\_social()



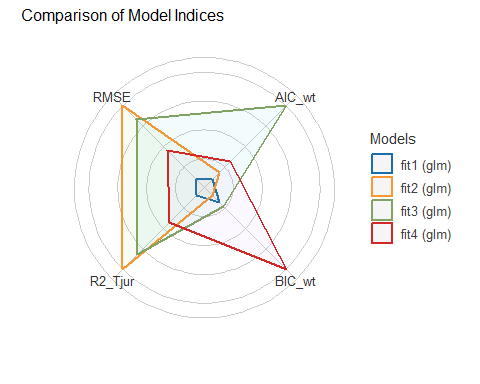
## Pseudo- measures (models so far)

r2\_res1234 <- tibble(name = c("fit1", "fit2", "fit3", "fit4"),   
 McFadden = c(as.numeric(r2\_mcfadden(fit1)[1]),  
 as.numeric(r2\_mcfadden(fit2)[1]),  
 as.numeric(r2\_mcfadden(fit3)[1]),  
 as.numeric(r2\_mcfadden(fit4)[1])),  
 Nagelkerke = c(as.numeric(r2\_nagelkerke(fit1)),  
 as.numeric(r2\_nagelkerke(fit2)),  
 as.numeric(r2\_nagelkerke(fit3)),  
 as.numeric(r2\_nagelkerke(fit4))),  
 Tjur = c(as.numeric(r2\_tjur(fit1)),  
 as.numeric(r2\_tjur(fit2)),  
 as.numeric(r2\_tjur(fit3)),  
 as.numeric(r2\_tjur(fit4))))  
  
r2\_res1234 |> gt() |> fmt\_number(decimals = 4) |>   
 tab\_options(table.font.size = 24) |>  
 opt\_stylize(style = 4, color = "green")

| name | McFadden | Nagelkerke | Tjur |
| --- | --- | --- | --- |
| fit1 | 0.1571 | 0.2590 | 0.1813 |
| fit2 | 0.2138 | 0.3398 | 0.2549 |
| fit3 | 0.2000 | 0.3206 | 0.2397 |
| fit4 | 0.1751 | 0.2853 | 0.2082 |

## Comparing Model Indices from our 4 models

plot(compare\_performance(fit1, fit2, fit3, fit4, metrics = "common"))



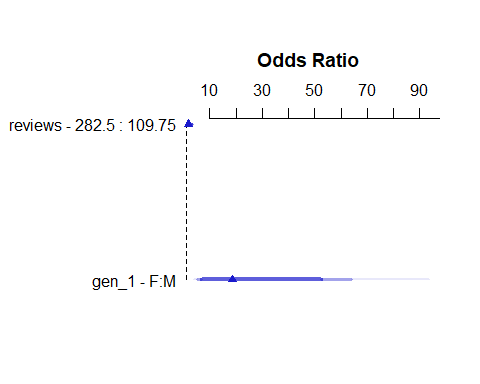
## What are the effect sizes in fit4\_lrm?

summary(fit4\_lrm)

Effects Response : bechdel   
  
 Factor Low High Diff. Effect S.E. Lower 0.95 Upper 0.95  
 reviews 109.75 282.5 172.75 0.67127 0.25089 0.17954 1.1630   
 Odds Ratio 109.75 282.5 172.75 1.95670 NA 1.19670 3.1995   
 gen\_1 - F:M 1.00 2.0 NA 2.92660 0.62676 1.69820 4.1551   
 Odds Ratio 1.00 2.0 NA 18.66500 NA 5.46420 63.7560

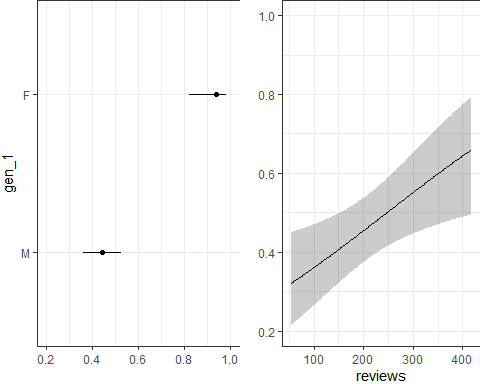
## Plotting the Effects for fit4\_lrm

plot(summary(fit4\_lrm))



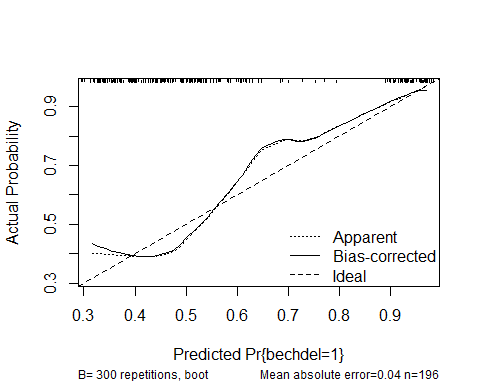
## Prediction Plots for fit4\_lrm

ggplot(Predict(fit4\_lrm, fun = plogis), layout = c(1,2))



## Calibration Plot for fit4\_lrm

set.seed(4321234)  
plot(calibrate(fit4\_lrm, method = "boot", B = 300))



n=196 Mean absolute error=0.04 Mean squared error=0.00248  
0.9 Quantile of absolute error=0.084

## fit4 Hosmer-Lemeshow test

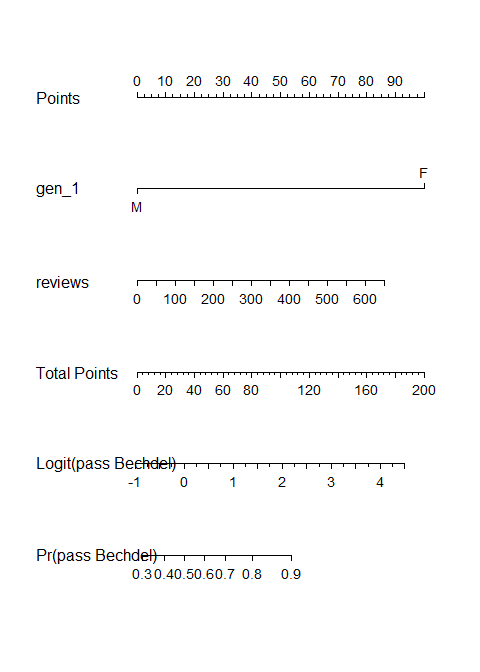
performance\_hosmer(fit4, n\_bins = 10)

# Hosmer-Lemeshow Goodness-of-Fit Test  
  
 Chi-squared: 11.284  
 df: 8   
 p-value: 0.186

Summary: model seems to fit well.

## Nomogram for fit4\_lrm

plot(nomogram(fit4\_lrm, fun = plogis, funlabel = "Pr(pass Bechdel)"),  
 lplabel = "Logit(pass Bechdel)")



## Predictions from fit4 for love4 movies

augment(fit4, newdata = love4, type.predict = "response") |>  
 select(movie, .fitted, bechdel, everything()) |>  
 gt() |> tab\_options(table.font.size = 20) |>  
 opt\_stylize(style = 5, color = "pink")

| movie | .fitted | bechdel | mov\_id | age | gen\_1 | mpa3 | reviews | ebert | romance | action |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The Manchurian Candidate | 0.3320302 | 0 | L-2 | 63 | M | PG-13 | 67 | 4.0 | 0 | 0 |
| Die Hard | 0.3512544 | 0 | L-8 | 37 | M | R | 89 | 2.0 | 0 | 1 |
| Love Actually | 0.4835973 | 1 | L-63 | 22 | M | Other | 230 | 3.5 | 1 | 0 |
| Sense and Sensibility | 0.9027032 | 1 | L-125 | 30 | F | Other | 67 | 3.5 | 1 | 0 |

## CIs around our predictions?

augment(fit4, newdata = love4, type.predict = "link", se\_fit = TRUE) |>  
 mutate(ci\_90\_low = .fitted - 1.645 \* .se.fit,   
 ci\_90\_high = .fitted + 1.645 \* .se.fit) |>  
 select(movie, .fitted, .se.fit, ci\_90\_low, ci\_90\_high, bechdel, everything())

# A tibble: 4 × 14  
 movie .fitted .se.fit ci\_90\_low ci\_90\_high bechdel mov\_id age gen\_1 mpa3   
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <fct> <fct>  
1 The Man… -0.699 0.265 -1.14 -0.263 0 L-2 63 M PG-13  
2 Die Hard -0.614 0.241 -1.01 -0.217 0 L-8 37 M R   
3 Love Ac… -0.0656 0.171 -0.347 0.216 1 L-63 22 M Other  
4 Sense a… 2.23 0.616 1.21 3.24 1 L-125 30 F Other  
# ℹ 4 more variables: reviews <dbl>, ebert <dbl>, romance <dbl>, action <dbl>

## Converting from Logit to Probability Scale

For Die Hard, our predicted logit(bechdel pass) = -0.6135, with 90% CI (-1.0104, -0.2167).

* If logit(bechdel pass) = -0.6135, then odds(bechdel pass) = exp(-0.6135), and pr(bechdel pass) = exp(-0.6135) / (1 + exp(-0.6135)) = 0.5415/1.5415 = 0.351
* If logit(bechdel pass) = -1.0104, then odds(bechdel pass) = exp(-1.0104), and pr(bechdel pass) = exp(-1.0104) / (1 + exp(-1.0104)) = 0.3641/1.3641 = 0.267
* If logit(bechdel pass) = -0.2167, then odds(bechdel pass) = exp(-0.2167), and pr(bechdel pass) = exp(-0.2167) / (1 + exp(-0.2167)) = 0.8052/1.8052 = 0.446

Predicted prob(bechdel pass) = 0.351 with 90% confidence interval (0.267, 0.446) for Die Hard using fit4.

## Picking a Decision Rule for fit4

* Again, using cutpointr to select a decision rule which maximizes “Sensitivity” + “Specificity”.

fit4\_aug <- augment(fit4, type.predict = "response")  
  
cp4 <- cutpointr(data = fit4\_aug, .fitted, bechdel,   
 pos\_class = 1, neg\_class = 0,  
 method = maximize\_metric, metric = sum\_sens\_spec)

Assuming the positive class has higher x values

cp4 |> select(direction, optimal\_cutpoint, method, sum\_sens\_spec) |>   
 gt() |> tab\_options(table.font.size = 24) |>   
 opt\_stylize(style = 2, color = "pink")

| direction | optimal\_cutpoint | method | sum\_sens\_spec |
| --- | --- | --- | --- |
| >= | 0.5847669 | maximize\_metric | 1.482034 |

## Confusion Matrix for fit4

cm4 <- confusionMatrix(data = factor(fit4\_aug$.fitted >= cp4$optimal\_cutpoint),  
 reference = factor(fit4\_aug$bechdel == 1), positive = "TRUE")  
cm4

Confusion Matrix and Statistics  
  
 Reference  
Prediction FALSE TRUE  
 FALSE 76 49  
 TRUE 7 64  
   
 Accuracy : 0.7143   
 95% CI : (0.6456, 0.7764)  
 No Information Rate : 0.5765   
 P-Value [Acc > NIR] : 4.645e-05   
   
 Kappa : 0.4517   
   
 Mcnemar's Test P-Value : 4.281e-08   
   
 Sensitivity : 0.5664   
 Specificity : 0.9157   
 Pos Pred Value : 0.9014   
 Neg Pred Value : 0.6080   
 Prevalence : 0.5765   
 Detection Rate : 0.3265   
 Detection Prevalence : 0.3622   
 Balanced Accuracy : 0.7410   
   
 'Positive' Class : TRUE

## fit4 PCP

Percentage of Correct Predictions (with 0.5 decision rule)

performance\_pcp(fit4, ci = 0.90, method = "Herron")

# Percentage of Correct Predictions from Logistic Regression Model  
  
 Full model: 61.34% [55.61% - 67.06%]  
 Null model: 51.17% [45.30% - 57.04%]  
  
# Likelihood-Ratio-Test  
  
 Chi-squared: 46.770  
 df: 2.000  
 p-value: 0.000

## No severe multicollinearity? (fit4)

car::vif(fit4)

gen\_1 reviews   
1.011406 1.011406

rms::vif(fit4\_lrm)

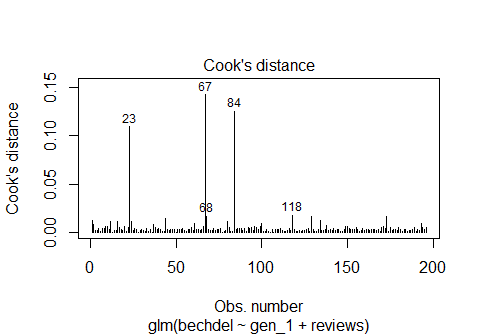
gen\_1=F reviews   
1.011406 1.011406

## No Cook’s distance > 0.5? (fit4)

max(cooks.distance(fit4))

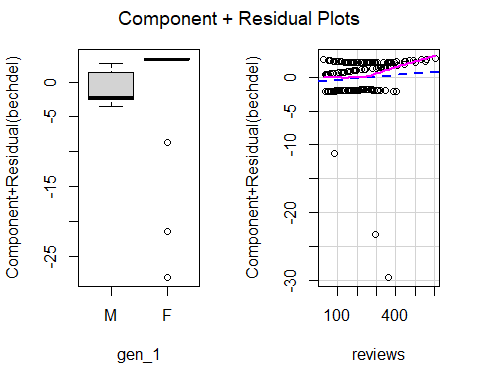
[1] 0.1424706

plot(fit4, which = 4, id.n = 5)



## fit4 Partial Residual Plots

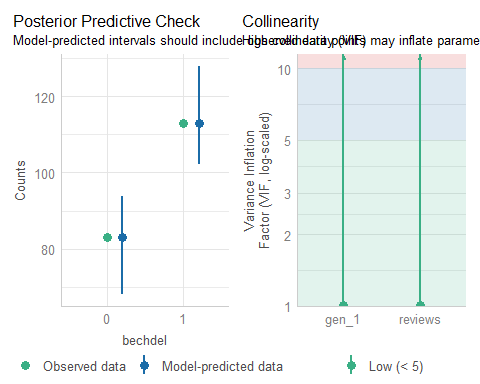
crPlots(fit4) ## crPlots comes from the car package



## check\_model for fit4 (1/4)

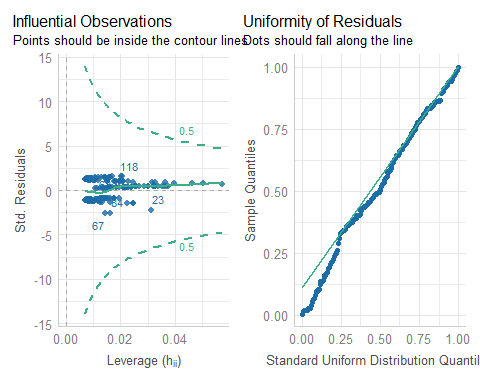
check\_model(fit4, check = c("pp\_check", "vif"))

Cannot simulate residuals for models of class `glm`. Please try  
 `check\_model(..., residual\_type = "normal")` instead.



## check\_model for fit4 (2/4)

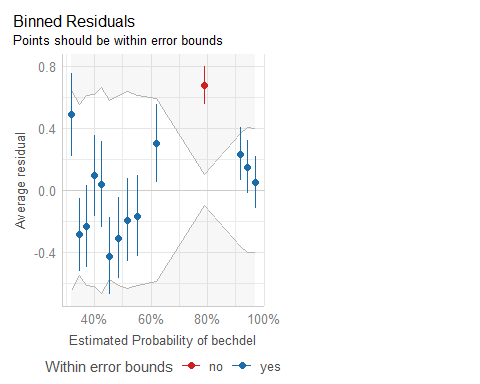
check\_model(fit4, check = c("outliers", "qq"))



## check\_model for fit4 (3/4)

check\_model(fit4, check = c("binned\_residuals"))

Cannot simulate residuals for models of class `glm`. Please try  
 `check\_model(..., residual\_type = "normal")` instead.



## check\_model for fit4 (4/4)

* Extra details for the last three plots…

check\_outliers(fit4)

OK: No outliers detected.  
- Based on the following method and threshold: cook (0.5).  
- For variable: (Whole model)

check\_residuals(fit4)

OK: Simulated residuals appear as uniformly distributed (p = 0.095).

binned\_residuals(fit4)

Warning: About 93% of the residuals are inside the error bounds (~95% or higher would be good).

## Analysis of Deviance for fit4

anova(fit4\_lrm)

Wald Statistics Response: bechdel   
  
 Factor Chi-Square d.f. P   
 gen\_1 21.80 1 <.0001  
 reviews 7.16 1 0.0075  
 TOTAL 26.61 2 <.0001

|  |
| --- |
| Note |
| Remember that this result shows sequential tests, and if you change the order of the predictors, the *p* values will change. |

## Validating Key Summaries (fit4)

set.seed(202502064); validate(fit4\_lrm, B = 300)

index.orig training test optimism index.corrected n  
Dxy 0.4862 0.4885 0.4845 0.0040 0.4822 300  
R2 0.2853 0.2942 0.2797 0.0144 0.2709 300  
Intercept 0.0000 0.0000 0.0098 -0.0098 0.0098 300  
Slope 1.0000 1.0000 0.9574 0.0426 0.9574 300  
Emax 0.0000 0.0000 0.0113 0.0113 0.0113 300  
D 0.2335 0.2437 0.2283 0.0154 0.2181 300  
U -0.0102 -0.0102 0.0239 -0.0341 0.0239 300  
Q 0.2437 0.2539 0.2043 0.0496 0.1942 300  
B 0.1924 0.1902 0.1952 -0.0050 0.1974 300  
g 1.4011 1.7127 1.3617 0.3510 1.0501 300  
gp 0.2455 0.2455 0.2389 0.0066 0.2389 300

* C = 0.5 + Dxy, so validated C for fit4 = 0.5 + (0.4822/2) = 0.7411, validated Nagelkerke = 0.2709, and validated Brier score B = 0.1974

## Cross-Validating AUC for fit4

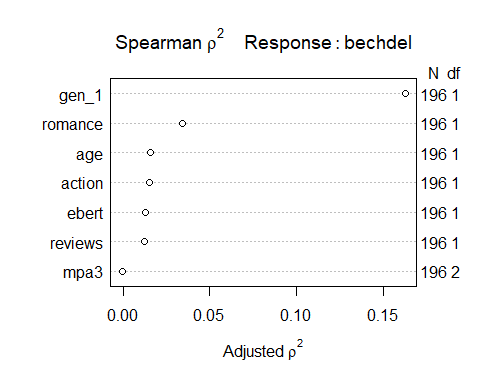
set.seed(123454)  
performance\_accuracy(fit4, method = "cv", k = 5, ci = 0.90)

# Accuracy of Model Predictions  
  
Accuracy (90% CI): 73.67% [62.18%, 88.94%]  
Method: Area under Curve

# Model fit5

## Spearman plot from fit2

plot(spearman2(bechdel ~ age + gen\_1 + mpa3 + reviews +   
 ebert + romance + action, data = mov25))



## Model fit5: add two non-linear terms

fit5 <- glm(bechdel ~ gen\_1 \* romance + rcs(age, 4) +   
 mpa3 + reviews + ebert + action,   
 data = mov25, family = binomial(link = "logit"))  
  
n\_obs(fit5)

[1] 196

performance\_roc(fit5)

AUC: 80.50%

model\_performance(fit5)

# Indices of model performance  
  
AIC | AICc | BIC | Tjur's R2 | RMSE | Sigma | Log\_loss | Score\_log  
------------------------------------------------------------------------------  
224.832 | 226.537 | 264.169 | 0.281 | 0.418 | 1.000 | 0.512 | -Inf  
  
AIC | Score\_spherical | PCP  
---------------------------------  
224.832 | 0.005 | 0.649

## Model fit5 parameters

model\_parameters(fit5, exponentiate = TRUE, ci = 0.90)

Parameter | Odds Ratio | SE | 90% CI | z | p  
--------------------------------------------------------------------------------  
(Intercept) | 0.69 | 0.91 | [0.08, 5.87] | -0.28 | 0.780   
gen 1 [F] | 8.21 | 5.57 | [2.94, 28.83] | 3.10 | 0.002   
romance | 1.36 | 0.70 | [0.59, 3.21] | 0.60 | 0.548   
rcs(age [ degree] | 0.94 | 0.08 | [0.81, 1.08] | -0.75 | 0.451   
rcs(age [ degree] | 2.31 | 1.12 | [1.06, 5.26] | 1.73 | 0.083   
rcs(age [ degree] | 0.14 | 0.15 | [0.02, 0.75] | -1.87 | 0.061   
mpa3 [R] | 0.68 | 0.31 | [0.32, 1.45] | -0.83 | 0.406   
mpa3 [Other] | 1.08 | 0.51 | [0.50, 2.37] | 0.17 | 0.864   
reviews | 1.01 | 2.84e-03 | [1.01, 1.02] | 3.64 | < .001  
ebert | 0.59 | 0.16 | [0.37, 0.91] | -1.93 | 0.053   
action | 0.40 | 0.18 | [0.19, 0.82] | -2.06 | 0.039   
gen 1 [F] × romance | 4.90e+06 | 3.97e+09 | [0.00, 1.84e+126] | 0.02 | 0.985

Uncertainty intervals (profile-likelihood) and p-values (two-tailed)  
 computed using a Wald z-distribution approximation.

## What is the fit5 equation?

fit5$coefficients ## note: without exponentiation

(Intercept) gen\_1F romance rcs(age, 4)age   
 -0.36512871 2.10491071 0.30870046 -0.06410644   
 rcs(age, 4)age' rcs(age, 4)age'' mpa3R mpa3Other   
 0.83895234 -1.94989001 -0.38266995 0.08117920   
 reviews ebert action gen\_1F:romance   
 0.01025840 -0.53201822 -0.92353679 15.40442177

No one writes an equation for a cubic spline fit. We use Prediction Plots and Nomograms from fit5\_lrm instead.

## lrm version of fit5

d <- datadist(mov25); options(datadist = "d")  
  
fit5\_lrm <- lrm(bechdel ~ gen\_1 \* romance + rcs(age, 4) +   
 mpa3 + reviews + ebert + action,  
 data = mov25, x = TRUE, y = TRUE)

|  |
| --- |
| Key Summaries for fit5\_lrm include… |
| * C = 0.805, Nagelkerke = 0.386, Brier score = 0.175 * See next slide for details. |

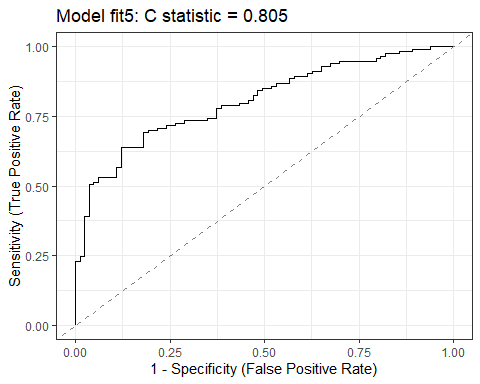
## fit5\_lrm summaries

fit5\_lrm

Logistic Regression Model  
  
lrm(formula = bechdel ~ gen\_1 \* romance + rcs(age, 4) + mpa3 +   
 reviews + ebert + action, data = mov25, x = TRUE, y = TRUE)  
  
 Model Likelihood Discrimination Rank Discrim.   
 Ratio Test Indexes Indexes   
Obs 196 LR chi2 66.27 R2 0.386 C 0.805   
 0 83 d.f. 11 R2(11,196)0.246 Dxy 0.610   
 1 113 Pr(> chi2) <0.0001 R2(11,143.6)0.320 gamma 0.610   
max |deriv| 9e-05 Brier 0.175 tau-a 0.299   
  
 Coef S.E. Wald Z Pr(>|Z|)  
Intercept -0.3651 1.3054 -0.28 0.7797   
gen\_1=F 2.1049 0.6792 3.10 0.0019   
romance 0.3087 0.5140 0.60 0.5481   
age -0.0641 0.0851 -0.75 0.4512   
age' 0.8390 0.4836 1.73 0.0827   
age'' -1.9499 1.0405 -1.87 0.0609   
mpa3=R -0.3827 0.4603 -0.83 0.4058   
mpa3=Other 0.0812 0.4733 0.17 0.8638   
reviews 0.0103 0.0028 3.64 0.0003   
ebert -0.5320 0.2753 -1.93 0.0533   
action -0.9235 0.4481 -2.06 0.0393   
gen\_1=F \* romance 16.1095 1899.5014 0.01 0.9932

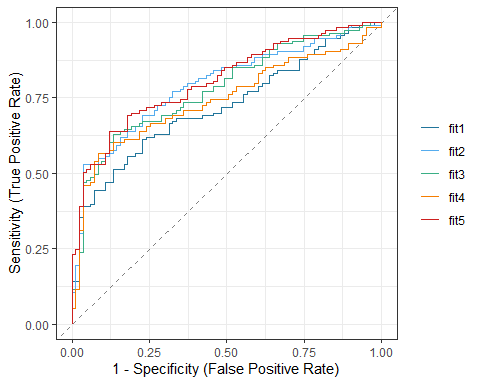
## ROC Curve Analysis for fit5

plot(performance\_roc(fit5)) +  
 labs(title = glue("Model fit5: C statistic = ",  
 round\_half\_up(as.numeric(performance\_roc(fit5)),3)))



## ROC curves for all 5 models

plot(performance\_roc(fit1, fit2, fit3, fit4, fit5)) +  
 scale\_color\_social()



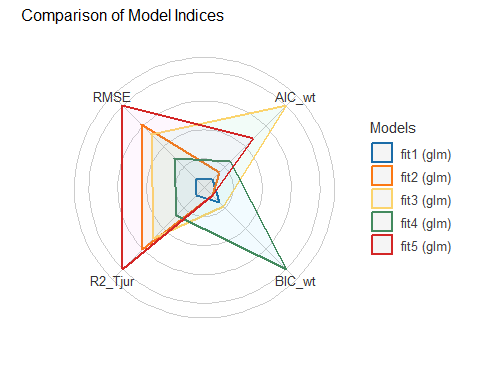
## Pseudo- measures (all 5 models)

r2\_res12345 <- tibble(name = c("fit1", "fit2", "fit3", "fit4", "fit5"),   
 McFadden = c(as.numeric(r2\_mcfadden(fit1)[1]),  
 as.numeric(r2\_mcfadden(fit2)[1]),  
 as.numeric(r2\_mcfadden(fit3)[1]),  
 as.numeric(r2\_mcfadden(fit4)[1]),  
 as.numeric(r2\_mcfadden(fit5)[1])),  
 Nagelkerke = c(as.numeric(r2\_nagelkerke(fit1)),  
 as.numeric(r2\_nagelkerke(fit2)),  
 as.numeric(r2\_nagelkerke(fit3)),  
 as.numeric(r2\_nagelkerke(fit4)),  
 as.numeric(r2\_nagelkerke(fit5))),  
 Tjur = c(as.numeric(r2\_tjur(fit1)),  
 as.numeric(r2\_tjur(fit2)),  
 as.numeric(r2\_tjur(fit3)),  
 as.numeric(r2\_tjur(fit4)),  
 as.numeric(r2\_tjur(fit5))))  
  
r2\_res12345 |> gt() |> fmt\_number(decimals = 4) |>   
 tab\_options(table.font.size = 24) |>  
 opt\_stylize(style = 4, color = "green")

| name | McFadden | Nagelkerke | Tjur |
| --- | --- | --- | --- |
| fit1 | 0.1571 | 0.2590 | 0.1813 |
| fit2 | 0.2138 | 0.3398 | 0.2549 |
| fit3 | 0.2000 | 0.3206 | 0.2397 |
| fit4 | 0.1751 | 0.2853 | 0.2082 |
| fit5 | 0.2481 | 0.3856 | 0.2814 |

## Comparing Model Indices from our 5 models

plot(compare\_performance(fit1, fit2, fit3, fit4, fit5, metrics = "common"))



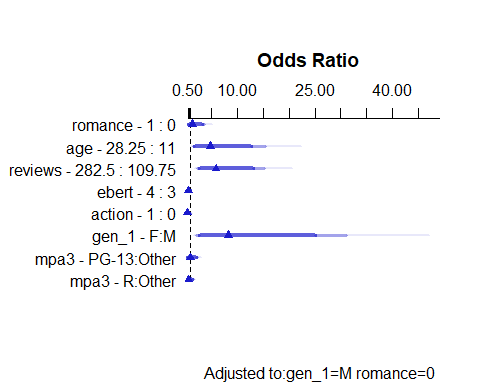
## What are the effect sizes in fit5\_lrm?

summary(fit5\_lrm)

Effects Response : bechdel   
  
 Factor Low High Diff. Effect S.E. Lower 0.95  
 romance 0.00 1.00 1.00 0.308700 0.51402 -0.69877   
 Odds Ratio 0.00 1.00 1.00 1.361700 NA 0.49720   
 age 11.00 28.25 17.25 1.556500 0.60354 0.37359   
 Odds Ratio 11.00 28.25 17.25 4.742200 NA 1.45290   
 reviews 109.75 282.50 172.75 1.772100 0.48642 0.81877   
 Odds Ratio 109.75 282.50 172.75 5.883400 NA 2.26770   
 ebert 3.00 4.00 1.00 -0.532020 0.27532 -1.07160   
 Odds Ratio 3.00 4.00 1.00 0.587420 NA 0.34245   
 action 0.00 1.00 1.00 -0.923540 0.44805 -1.80170   
 Odds Ratio 0.00 1.00 1.00 0.397110 NA 0.16502   
 gen\_1 - F:M 1.00 2.00 NA 2.104900 0.67918 0.77375   
 Odds Ratio 1.00 2.00 NA 8.206400 NA 2.16790   
 mpa3 - PG-13:Other 3.00 1.00 NA -0.081179 0.47328 -1.00880   
 Odds Ratio 3.00 1.00 NA 0.922030 NA 0.36466   
 mpa3 - R:Other 3.00 2.00 NA -0.463850 0.44262 -1.33140   
 Odds Ratio 3.00 2.00 NA 0.628860 NA 0.26412   
 Upper 0.95  
 1.3162000  
 3.7291000  
 2.7394000  
 15.4780000  
 2.7255000  
 15.2640000  
 0.0075955  
 1.0076000  
 -0.0453730  
 0.9556400  
 3.4361000  
 31.0650000  
 0.8464400  
 2.3313000  
 0.4036700  
 1.4973000  
  
Adjusted to: gen\_1=M romance=0

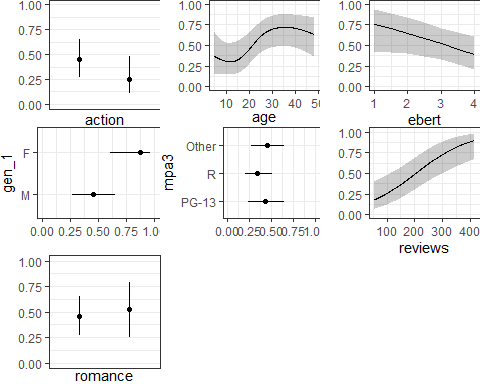
## Plotting the Effects for fit5\_lrm

plot(summary(fit5\_lrm))



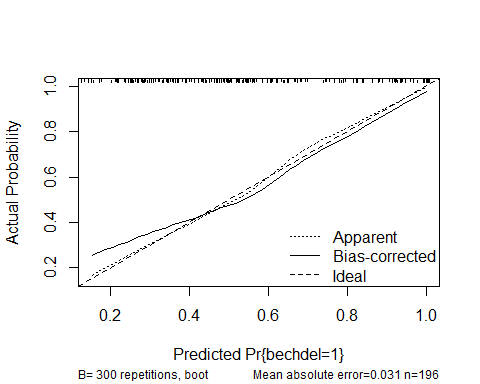
## Prediction Plots for fit5\_lrm

ggplot(Predict(fit5\_lrm, fun = plogis))



## Calibration Plot for fit5\_lrm

set.seed(4321235)  
plot(calibrate(fit5\_lrm, method = "boot", B = 300))



n=196 Mean absolute error=0.031 Mean squared error=0.00139  
0.9 Quantile of absolute error=0.061

## fit5 Hosmer-Lemeshow test

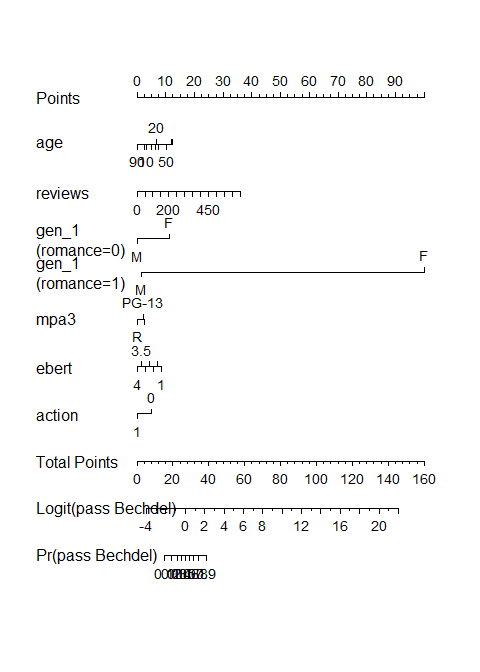
performance\_hosmer(fit5, n\_bins = 10)

# Hosmer-Lemeshow Goodness-of-Fit Test  
  
 Chi-squared: 6.273  
 df: 8   
 p-value: 0.617

Summary: model seems to fit well.

## Nomogram for fit5\_lrm

plot(nomogram(fit5\_lrm, fun = plogis, funlabel = "Pr(pass Bechdel)"),  
 lplabel = "Logit(pass Bechdel)")



## Predictions from fit5 for love4 movies

augment(fit5, newdata = love4, type.predict = "response") |>  
 select(movie, .fitted, bechdel, everything()) |>  
 gt() |> tab\_options(table.font.size = 20) |>  
 opt\_stylize(style = 5, color = "pink")

| movie | .fitted | bechdel | mov\_id | age | gen\_1 | mpa3 | reviews | ebert | romance | action |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The Manchurian Candidate | 0.1523043 | 0 | L-2 | 63 | M | PG-13 | 67 | 4.0 | 0 | 0 |
| Die Hard | 0.3427204 | 0 | L-8 | 37 | M | R | 89 | 2.0 | 0 | 1 |
| Love Actually | 0.6984687 | 1 | L-63 | 22 | M | Other | 230 | 3.5 | 1 | 0 |
| Sense and Sensibility | 1.0000000 | 1 | L-125 | 30 | F | Other | 67 | 3.5 | 1 | 0 |

## CIs around our predictions?

augment(fit5, newdata = love4, type.predict = "link", se\_fit = TRUE) |>  
 mutate(ci\_90\_low = .fitted - 1.645 \* .se.fit,   
 ci\_90\_high = .fitted + 1.645 \* .se.fit) |>  
 select(movie, .fitted, .se.fit, ci\_90\_low, ci\_90\_high, bechdel, everything())

# A tibble: 4 × 14  
 movie .fitted .se.fit ci\_90\_low ci\_90\_high bechdel mov\_id age gen\_1 mpa3   
 <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <fct> <fct>  
1 The Man… -1.72 0.992 -3.35 -0.0855 0 L-2 63 M PG-13  
2 Die Hard -0.651 0.640 -1.70 0.402 0 L-8 37 M R   
3 Love Ac… 0.840 0.623 -0.185 1.87 1 L-63 22 M Other  
4 Sense a… 17.4 810. -1315. 1350. 1 L-125 30 F Other  
# ℹ 4 more variables: reviews <dbl>, ebert <dbl>, romance <dbl>, action <dbl>

## Converting from Logit to Probability Scale

For Love Actually, our predicted logit(bechdel pass) = 0.8400, with 90% CI (-0.1852, 1.8653).

* If logit(bechdel pass) = 0.8400, then odds(bechdel pass) = exp(0.8400), and pr(bechdel pass) = exp(0.8400) / (1 + exp(0.8400)) = 2.3164/3.3164 = 0.698
* If logit(bechdel pass) = -0.1852, then odds(bechdel pass) = exp(-0.1852), and pr(bechdel pass) = exp(-0.1852) / (1 + exp(-0.1852)) = 0.8309/1.8309 = 0.454
* If logit(bechdel pass) = 1.8653, then odds(bechdel pass) = exp(1.8653), and pr(bechdel pass) = exp(1.8653) / (1 + exp(1.8653)) = 6.4578/7.4578 = 0.866

Predicted prob(bechdel pass) = 0.698 with 90% confidence interval (0.454, 0.866) for Love Actually using fit5.

## Picking a Decision Rule for fit5

* Again, using cutpointr to select a decision rule which maximizes “Sensitivity” + “Specificity”.

fit5\_aug <- augment(fit5, type.predict = "response")  
  
cp5 <- cutpointr(data = fit5\_aug, .fitted, bechdel,   
 pos\_class = 1, neg\_class = 0,  
 method = maximize\_metric, metric = sum\_sens\_spec)

Assuming the positive class has higher x values

cp5 |> select(direction, optimal\_cutpoint, method, sum\_sens\_spec) |>   
 gt() |> tab\_options(table.font.size = 24) |>   
 opt\_stylize(style = 2, color = "pink")

| direction | optimal\_cutpoint | method | sum\_sens\_spec |
| --- | --- | --- | --- |
| >= | 0.5626889 | maximize\_metric | 1.516686 |

## Confusion Matrix for fit5

cm5 <- confusionMatrix(data = factor(fit5\_aug$.fitted >= cp5$optimal\_cutpoint),  
 reference = factor(fit5\_aug$bechdel == 1), positive = "TRUE")  
cm5

Confusion Matrix and Statistics  
  
 Reference  
Prediction FALSE TRUE  
 FALSE 73 41  
 TRUE 10 72  
   
 Accuracy : 0.7398   
 95% CI : (0.6725, 0.7997)  
 No Information Rate : 0.5765   
 P-Value [Acc > NIR] : 1.474e-06   
   
 Kappa : 0.4923   
   
 Mcnemar's Test P-Value : 2.659e-05   
   
 Sensitivity : 0.6372   
 Specificity : 0.8795   
 Pos Pred Value : 0.8780   
 Neg Pred Value : 0.6404   
 Prevalence : 0.5765   
 Detection Rate : 0.3673   
 Detection Prevalence : 0.4184   
 Balanced Accuracy : 0.7583   
   
 'Positive' Class : TRUE

## fit5 PCP

Percentage of Correct Predictions (with 0.5 decision rule)

performance\_pcp(fit5, ci = 0.90, method = "Herron")

# Percentage of Correct Predictions from Logistic Regression Model  
  
 Full model: 64.91% [59.30% - 70.52%]  
 Null model: 51.17% [45.30% - 57.04%]  
  
# Likelihood-Ratio-Test  
  
 Chi-squared: 66.272  
 df: 11.000  
 p-value: 0.000

## No severe multicollinearity? (fit5)

car::vif(fit5)

there are higher-order terms (interactions) in this model  
consider setting type = 'predictor'; see ?vif

GVIF Df GVIF^(1/(2\*Df))  
gen\_1 1.032243 1 1.015994  
romance 1.121039 1 1.058792  
rcs(age, 4) 2.967515 3 1.198760  
mpa3 1.452883 2 1.097887  
reviews 3.115263 1 1.765011  
ebert 1.271118 1 1.127439  
action 1.432239 1 1.196762  
gen\_1:romance 1.000001 1 1.000001

rms::vif(fit5\_lrm)

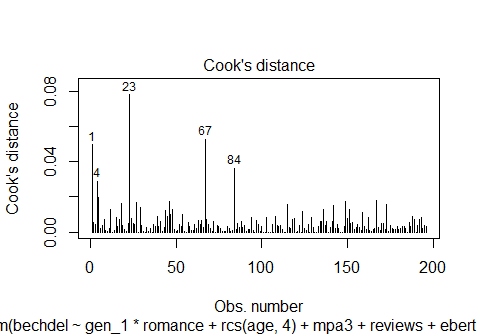
gen\_1=F romance age age'   
 1.032243 1.121039 43.630373 923.944699   
 age'' mpa3=R mpa3=Other reviews   
 617.481842 1.674308 1.691158 3.115263   
 ebert action gen\_1=F \* romance   
 1.271118 1.432239 1.000000

## No Cook’s distance > 0.5? (fit5)

max(cooks.distance(fit5))

[1] 0.07793543

plot(fit5, which = 4, id.n = 5)



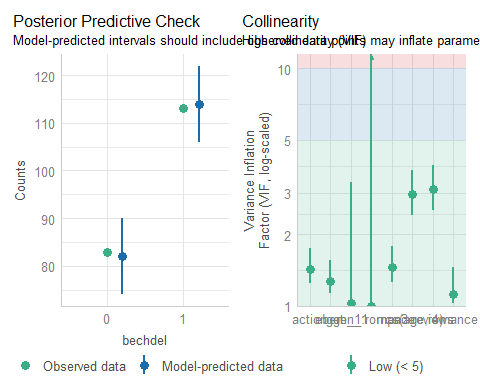
## fit5 Partial Residual Plots

Component + Residual plots are not available for models with interactions

## check\_model for fit5 (1/4)

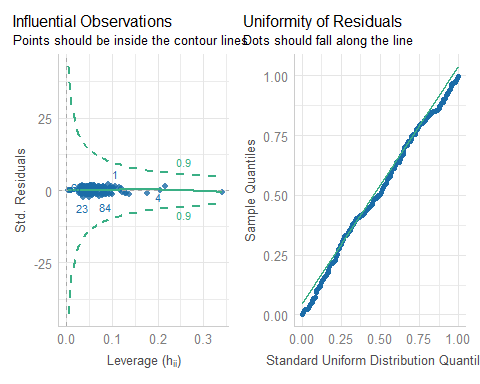
check\_model(fit5, check = c("pp\_check", "vif"))

Cannot simulate residuals for models of class `glm`. Please try  
 `check\_model(..., residual\_type = "normal")` instead.



## check\_model for fit5 (2/4)

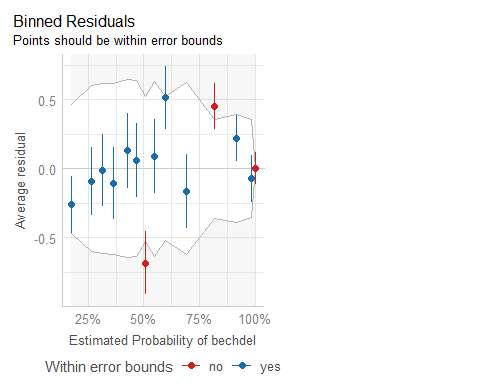
check\_model(fit5, check = c("outliers", "qq"))



## check\_model for fit5 (3/4)

check\_model(fit5, check = c("binned\_residuals"))

Cannot simulate residuals for models of class `glm`. Please try  
 `check\_model(..., residual\_type = "normal")` instead.



## check\_model for fit5 (4/4)

* Extra details for the last three plots…

check\_outliers(fit5)

OK: No outliers detected.  
- Based on the following method and threshold: cook (0.9).  
- For variable: (Whole model)

check\_residuals(fit5)

OK: Simulated residuals appear as uniformly distributed (p = 0.398).

binned\_residuals(fit5)

Warning: Probably bad model fit. Only about 79% of the residuals are inside the error bounds.

## Analysis of Deviance for fit5

anova(fit5\_lrm)

Wald Statistics Response: bechdel   
  
 Factor Chi-Square d.f. P   
 gen\_1 (Factor+Higher Order Factors) 9.61 2 0.0082  
 All Interactions 0.00 1 0.9932  
 romance (Factor+Higher Order Factors) 0.36 2 0.8350  
 All Interactions 0.00 1 0.9932  
 age 6.83 3 0.0776  
 Nonlinear 5.57 2 0.0618  
 mpa3 1.26 2 0.5327  
 reviews 13.27 1 0.0003  
 ebert 3.73 1 0.0533  
 action 4.25 1 0.0393  
 gen\_1 \* romance (Factor+Higher Order Factors) 0.00 1 0.9932  
 TOTAL NONLINEAR + INTERACTION 5.57 3 0.1347  
 TOTAL 26.50 11 0.0055

## Validating Key Summaries (fit5)

set.seed(202502065); validate(fit5\_lrm, B = 300)

index.orig training test optimism index.corrected n  
Dxy 0.6100 0.6602 0.5700 0.0903 0.5197 300  
R2 0.3856 0.4383 0.3433 0.0950 0.2906 300  
Intercept 0.0000 0.0000 -0.0097 0.0097 -0.0097 300  
Slope 1.0000 1.0000 0.6926 0.3074 0.6926 300  
Emax 0.0000 0.0000 0.0831 0.0831 0.0831 300  
D 0.3330 0.3909 0.2901 0.1007 0.2323 300  
U -0.0102 -0.0102 0.0454 -0.0556 0.0454 300  
Q 0.3432 0.4011 0.2448 0.1563 0.1869 300  
B 0.1749 0.1628 0.1872 -0.0244 0.1993 300  
g 4.6196 5.1042 3.4482 1.6560 2.9636 300  
gp 0.2990 0.3224 0.2759 0.0465 0.2525 300

* C = 0.5 + Dxy, so validated C for fit5 = 0.5 + (0.5197/2) = 0.7599, validated Nagelkerke = 0.2906, and validated Brier score B = 0.1993

## Cross-Validating AUC for fit5

set.seed(123455)  
performance\_accuracy(fit5, method = "cv", k = 5, ci = 0.90)

# Accuracy of Model Predictions  
  
Accuracy (90% CI): 72.42% [67.26%, 79.23%]  
Method: Area under Curve

# Comparing Our Five Models

## compare\_parameters() results (1 and 2)

compare\_parameters(fit1, fit2, ci = 0.90)

Parameter | fit1 | fit2  
------------------------------------------------------------  
(Intercept) | 0.29 (-0.22, 0.79) | -0.05 (-1.61, 1.50)  
age | -0.02 (-0.04, 0.00) | 0.02 (-0.01, 0.04)  
gen 1 [F] | 2.81 ( 1.79, 3.83) | 2.59 ( 1.53, 3.64)  
mpa3 [Other] | | 0.10 (-0.65, 0.85)  
reviews | | 7.15e-03 ( 0.00, 0.01)  
ebert | | -0.53 (-0.98, -0.08)  
mpa3 [R] | | -0.30 (-1.04, 0.45)  
action | | -0.77 (-1.47, -0.06)  
romance | | 0.52 (-0.25, 1.29)  
------------------------------------------------------------  
Observations | 196 | 196

## compare\_parameters() results (3 and 4)

compare\_parameters(fit3, fit4, ci = 0.90)

Parameter | fit3 | fit4  
----------------------------------------------------------------  
(Intercept) | 0.53 (-0.83, 1.89) | -0.96 (-1.53, -0.39)  
gen 1 [F] | 2.71 ( 1.67, 3.75) | 2.93 ( 1.90, 3.96)  
reviews | 5.14e-03 ( 0.00, 0.01) | 3.89e-03 ( 0.00, 0.01)  
ebert | -0.47 (-0.88, -0.05) |   
action | -0.72 (-1.34, -0.09) |   
----------------------------------------------------------------  
Observations | 196 | 196

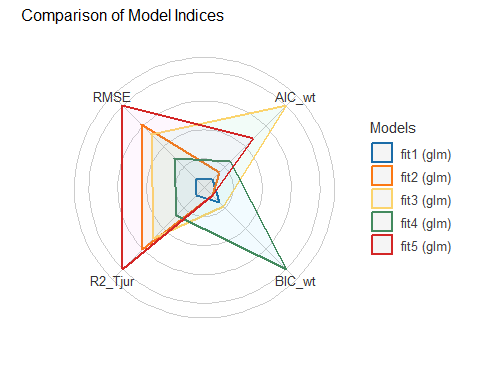
## compare\_parameters() results (2 and 5)

compare\_parameters(fit2, fit5, ci = 0.90)

Parameter | fit2 | fit5  
-------------------------------------------------------------------------  
(Intercept) | -0.05 (-1.61, 1.50) | -0.37 ( -2.51, 1.78)  
gen 1 [F] | 2.59 ( 1.53, 3.64) | 2.10 ( 0.99, 3.22)  
mpa3 [R] | -0.30 (-1.04, 0.45) | -0.38 ( -1.14, 0.37)  
mpa3 [Other] | 0.10 (-0.65, 0.85) | 0.08 ( -0.70, 0.86)  
reviews | 7.15e-03 ( 0.00, 0.01) | 0.01 ( 0.01, 0.01)  
ebert | -0.53 (-0.98, -0.08) | -0.53 ( -0.98, -0.08)  
romance | 0.52 (-0.25, 1.29) | 0.31 ( -0.54, 1.15)  
action | -0.77 (-1.47, -0.06) | -0.92 ( -1.66, -0.19)  
age | 0.02 (-0.01, 0.04) |   
rcs(age [ degree] | | 0.84 ( 0.04, 1.63)  
rcs(age [ degree] | | -1.95 ( -3.66, -0.24)  
rcs(age [ degree] | | -0.06 ( -0.20, 0.08)  
gen 1 [F] × romance | | 15.40 (-1316.65, 1347.46)  
-------------------------------------------------------------------------  
Observations | 196 | 196

## compare\_performance() results

plot(compare\_performance(fit1, fit2, fit3, fit4, fit5, metrics = "common"))



## Summary Statistics

| Model | fit1 | fit2 | fit3 | fit4 | fit5 |
| --- | --- | --- | --- | --- | --- |
| Model df | 2 | 8 | 4 | 2 | 11 |
| C (unvalidated) | 0.7199 | 0.7921 | 0.7792 | 0.7430 | 0.8050 |
| C (validated) | 0.7183 | 0.7581 | 0.7670 | 0.7411 | 0.7599 |
| Nagelkerke (un.) | 0.2590 | 0.3398 | 0.3206 | 0.2853 | 0.3856 |
| Nagelkerke (val.) | 0.2469 | 0.2674 | 0.2905 | 0.2709 | 0.2906 |
| Tjur (unval.) | 0.181 | 0.255 | 0.240 | 0.208 | 0.281 |
| McFadden (un.) | 0.157 | 0.213 | 0.200 | 0.175 | 0.248 |

## Confusion Matrix Summaries

| Model | fit1 | fit2 | fit3 | fit4 | fit5 |
| --- | --- | --- | --- | --- | --- |
| Cutpoint | 0.533 | 0.695 | 0.563 | 0.584 | 0.563 |
| Accuracy | 0.663 | 0.714 | 0.730 | 0.714 | 0.740 |
| Sensitivity | 0.513 | 0.531 | 0.628 | 0.566 | 0.637 |
| Specificity | 0.868 | 0.964 | 0.868 | 0.916 | 0.880 |
| Pos Pre Val | 0.841 | 0.952 | 0.866 | 0.901 | 0.878 |
| Neg Pre Val | 0.567 | 0.602 | 0.632 | 0.608 | 0.640 |
| Kappa | 0.356 | 0.458 | 0.472 | 0.452 | 0.492 |

## More Model Summaries

| Model | fit1 | fit2 | fit3 | fit4 | fit5 |
| --- | --- | --- | --- | --- | --- |
| Mean absolute error | 0.018 | 0.026 | 0.018 | 0.040 | 0.031 |
| Mean squared error | 0.00046 | 0.00112 | 0.00052 | 0.00248 | 0.00139 |
| 0.9 quant abs error | 0.036 | 0.053 | 0.033 | 0.084 | 0.061 |
| Hosmer-Lemeshow *p* | 0.747 | 0.358 | 0.265 | 0.186 | 0.617 |
| Brier (unval.) | 0.1998 | 0.1811 | 0.1845 | 0.1924 | 0.1749 |
| Validated Brier | 0.2042 | 0.1989 | 0.1927 | 0.1974 | 0.1993 |
| % Correct Preds. | 0.6002 | 0.6362 | 0.6288 | 0.6134 | 0.6491 |
| Cross-Val AUC | 0.7293 | 0.7594 | 0.7433 | 0.7367 | 0.7242 |

## Predictions for love4 movies

| Movie | bechdel | fit1 | fit2 | fit3 | fit4 | fit5 |
| --- | --- | --- | --- | --- | --- | --- |
| The Manchurian Candidate | 0 | 0.283 | 0.344 | 0.271 | 0.332 | 0.152 |
| Die Hard | 0 | 0.394 | 0.284 | 0.341 | 0.351 | 0.343 |
| Love Actually | 1 | 0.465 | 0.675 | 0.520 | 0.484 | 0.698 |
| Sense & Sensibility | 1 | 0.925 | 0.908 | 0.876 | 0.903 | 1.000 |

## What’s not in this example?

* Missing data and single/multiple imputation with mice or aregImpute()
* Partitioning the data into training and test samples
* Evaluation of a logistic regression model in a test sample
* Fitting a Bayesian logistic regression model

The support1000 example covers each of these pieces, in addition to the majority of the material in this mov25 example.

## Session Information

xfun::session\_info()

R version 4.4.2 (2024-10-31 ucrt)  
Platform: x86\_64-w64-mingw32/x64  
Running under: Windows 11 x64 (build 22631)  
  
Locale:  
 LC\_COLLATE=English\_United States.utf8   
 LC\_CTYPE=English\_United States.utf8   
 LC\_MONETARY=English\_United States.utf8  
 LC\_NUMERIC=C   
 LC\_TIME=English\_United States.utf8   
  
Package version:  
 abind\_1.4-8 ape\_5.8.1 askpass\_1.2.1   
 backports\_1.5.0 base64enc\_0.1-3 bayestestR\_0.15.1   
 bestglm\_0.37.3 bigD\_0.3.0 bit\_4.5.0.1   
 bit64\_4.6.0.1 bitops\_1.0.9 blob\_1.2.4   
 boot\_1.3-31 broom\_1.0.7 bslib\_0.9.0   
 cachem\_1.1.0 callr\_3.7.6 car\_3.1-3   
 carData\_3.0-5 caret\_7.0-1 cellranger\_1.1.0   
 checkmate\_2.3.2 chk\_0.10.0 class\_7.3-22   
 cli\_3.6.3 clipr\_0.8.0 clock\_0.7.2   
 cluster\_2.1.6 cobalt\_4.5.5 coda\_0.19-4.1   
 codetools\_0.2-20 colorspace\_2.1-1 commonmark\_1.9.2   
 compiler\_4.4.2 conflicted\_1.2.0 correlation\_0.8.6   
 cowplot\_1.1.3 cpp11\_0.5.1 crayon\_1.5.3   
 crosstalk\_1.2.1 curl\_6.2.0 cutpointr\_1.2.0   
 data.table\_1.16.4 datasets\_4.4.2 datawizard\_1.0.0   
 DBI\_1.2.3 dbplyr\_2.5.0 Deriv\_4.1.6   
 DHARMa\_0.4.7 diagram\_1.6.5 digest\_0.6.37   
 doBy\_4.6.25 doParallel\_1.0.17 dplyr\_1.1.4   
 dtplyr\_1.3.1 e1071\_1.7-16 easystats\_0.7.3   
 effectsize\_1.0.0 emmeans\_1.10.7 estimability\_1.5.1   
 evaluate\_1.0.3 fansi\_1.0.6 farver\_2.1.2   
 fastmap\_1.2.0 fontawesome\_0.5.3 forcats\_1.0.0   
 foreach\_1.5.2 foreign\_0.8-88 Formula\_1.2-5   
 fs\_1.6.5 future\_1.34.0 future.apply\_1.11.3   
 gap\_1.6 gap.datasets\_0.0.6 gargle\_1.5.2   
 gdata\_3.0.1 generics\_0.1.3 ggplot2\_3.5.1   
 ggrepel\_0.9.6 glmnet\_4.1-8 globals\_0.16.3   
 glue\_1.8.0 gmodels\_2.19.1 googledrive\_2.1.1   
 googlesheets4\_1.1.1 gower\_1.0.2 graphics\_4.4.2   
 grDevices\_4.4.2 grid\_4.4.2 gridExtra\_2.3   
 grpreg\_3.5.0 gt\_0.11.1 gtable\_0.3.6   
 gtools\_3.9.5 hardhat\_1.4.1 haven\_2.5.4   
 highr\_0.11 Hmisc\_5.2-2 hms\_1.1.3   
 htmlTable\_2.4.3 htmltools\_0.5.8.1 htmlwidgets\_1.6.4   
 httpuv\_1.6.15 httr\_1.4.7 ids\_1.0.1   
 insight\_1.0.1 ipred\_0.9-15 isoband\_0.2.7   
 iterators\_1.0.14 janitor\_2.2.1 jquerylib\_0.1.4   
 jsonlite\_1.8.9 juicyjuice\_0.1.0 KernSmooth\_2.23.24   
 knitr\_1.49 labeling\_0.4.3 labelled\_2.14.0   
 later\_1.4.1 lattice\_0.22-6 lava\_1.8.1   
 lazyeval\_0.2.2 leaps\_3.2 lifecycle\_1.0.4   
 listenv\_0.9.1 lme4\_1.1-36 lmtest\_0.9.40   
 lubridate\_1.9.4 magrittr\_2.0.3 markdown\_1.13   
 MASS\_7.3-64 Matrix\_1.7-1 MatrixModels\_0.5-3   
 memoise\_2.0.1 methods\_4.4.2 mgcv\_1.9-1   
 microbenchmark\_1.5.0 mime\_0.12 minqa\_1.2.8   
 mitools\_2.4 modelbased\_0.8.9 ModelMetrics\_1.2.2.2   
 modelr\_0.1.11 multcomp\_1.4-28 munsell\_0.5.1   
 mvtnorm\_1.3-3 naniar\_1.1.0 nlme\_3.1-166   
 nloptr\_2.1.1 nnet\_7.3-20 norm\_1.0.11.1   
 numDeriv\_2016.8.1.1 openssl\_2.3.2 parallel\_4.4.2   
 parallelly\_1.42.0 parameters\_0.24.1 patchwork\_1.3.0   
 pbkrtest\_0.5.3 performance\_0.13.0 pillar\_1.10.1   
 pkgconfig\_2.0.3 plotly\_4.10.4 pls\_2.8-5   
 plyr\_1.8.9 polspline\_1.1.25 prettyunits\_1.2.0   
 pROC\_1.18.5 processx\_3.8.5 prodlim\_2024.06.25   
 progress\_1.2.3 progressr\_0.15.1 promises\_1.3.2   
 proxy\_0.4-27 ps\_1.8.1 purrr\_1.0.2   
 qgam\_1.3.4 quantreg\_6.00 R6\_2.5.1   
 ragg\_1.3.3 rappdirs\_0.3.3 rbibutils\_2.3   
 RColorBrewer\_1.1.3 Rcpp\_1.0.14 RcppArmadillo\_14.2.2.1  
 RcppEigen\_0.3.4.0.2 Rdpack\_2.6.2 reactable\_0.4.4   
 reactR\_0.6.1 readr\_2.1.5 readxl\_1.4.3   
 recipes\_1.1.0 reformulas\_0.4.0 rematch\_2.0.0   
 rematch2\_2.1.2 report\_0.6.0 reprex\_2.1.1   
 reshape2\_1.4.4 rlang\_1.1.5 rmarkdown\_2.29   
 rms\_7.0-0 rpart\_4.1.24 rstudioapi\_0.17.1   
 rvest\_1.0.4 sandwich\_3.1-1 sass\_0.4.9   
 scales\_1.3.0 see\_0.10.0 selectr\_0.4.2   
 shape\_1.4.6.1 shiny\_1.10.0 snakecase\_0.11.1   
 sourcetools\_0.1.7.1 SparseM\_1.84-2 sparsevctrs\_0.2.0   
 splines\_4.4.2 SQUAREM\_2021.1 stats\_4.4.2   
 stats4\_4.4.2 stringi\_1.8.4 stringr\_1.5.1   
 survey\_4.4-2 survival\_3.8-3 sys\_3.4.3   
 systemfonts\_1.2.1 tableone\_0.13.2 textshaping\_1.0.0   
 TH.data\_1.1-3 tibble\_3.2.1 tidyr\_1.3.1   
 tidyselect\_1.2.1 tidyverse\_2.0.0 timechange\_0.3.0   
 timeDate\_4041.110 tinytex\_0.54 tools\_4.4.2   
 tzdb\_0.4.0 UpSetR\_1.4.0 utf8\_1.2.4   
 utils\_4.4.2 uuid\_1.2.1 V8\_6.0.1   
 vctrs\_0.6.5 viridis\_0.6.5 viridisLite\_0.4.2   
 visdat\_0.6.0 vroom\_1.6.5 withr\_3.0.2   
 xfun\_0.50 xml2\_1.3.6 xtable\_1.8-4   
 yaml\_2.3.10 zoo\_1.8-12

1. See <https://feministfrequency.com/video/the-bechdel-test-for-women-in-movies/> [↑](#footnote-ref-31)
2. Some argue for a standard of 0.25, or 1, or even 4/n, where n is your sample size. [↑](#footnote-ref-158)
3. Fox, J. (2016) Applied Regression Analysis and Generalized Linear Models, 3rd Ed. [↑](#footnote-ref-175)