431 Class 20

https://thomaselove.github.io/431-2024/

2024-11-07

## Today’s Agenda

1. Reviewing What We Did Last Week
2. Building Three Candidate Prediction Models
   * Assessing coefficients (model parameters)
   * Obtaining summaries of fit quality (performance)
3. Comparing some in-sample performance indices.
4. Checking model assumptions with check\_model() in the training sample

* Thinking more deeply about predictions and residuals
* Collinearity (correlated predictors)

## 431 strategy: “most useful” model?

1. Split the data into a development (model training) sample of about 70-80% of the observations, and a holdout (model test) sample, containing the remaining observations.
2. Develop candidate models using the development sample.
3. Assess the quality of fit for candidate models within the development sample.

## 431 strategy: “most useful” model?

1. Check adherence to regression assumptions in the development sample.
2. When you have candidates, assess them based on the accuracy of the predictions they make for the data held out (and thus not used in building the models.)
3. Select a “final” model for use based on the evidence in steps 3, 4 and especially 5.

## R Packages

knitr::opts\_chunk$set(comment = NA)  
  
library(janitor)

Attaching package: 'janitor'

The following objects are masked from 'package:stats':  
  
 chisq.test, fisher.test

library(mice)

Attaching package: 'mice'

The following object is masked from 'package:stats':  
  
 filter

The following objects are masked from 'package:base':  
  
 cbind, rbind

library(naniar)  
library(patchwork)  
library(car) ## for vif function as well as Box-Cox

Loading required package: carData

library(GGally) ## for ggpairs scatterplot matrix

Loading required package: ggplot2

Registered S3 method overwritten by 'GGally':  
 method from   
 +.gg ggplot2

library(broom) ## for predictions, residuals with augment  
library(easystats)

# Attaching packages: easystats 0.7.3  
✔ bayestestR 0.15.0 ✔ correlation 0.8.6   
✔ datawizard 0.13.0 ✔ effectsize 0.8.9   
✔ insight 0.20.5 ✔ modelbased 0.8.9   
✔ performance 0.12.4 ✔ parameters 0.23.0  
✔ report 0.5.9 ✔ see 0.9.0

library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.5  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ lubridate 1.9.3 ✔ tibble 3.2.1  
✔ purrr 1.0.2 ✔ tidyr 1.3.1

── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks mice::filter(), stats::filter()  
✖ dplyr::lag() masks stats::lag()  
✖ dplyr::recode() masks car::recode()  
✖ purrr::some() masks car::some()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

source("c20/data/Love-431.R")  
  
theme\_set(theme\_bw())

## Today’s Data

The dm500.Rds data contains four important variables + Subject ID on 500 adults with diabetes.

We want to predict the subject’s current Hemoglobin A1c level (a1c), using (up to) three predictors:

* a1c\_old: subject’s Hemoglobin A1c (in %) two years ago
* age: subject’s age in years (between 30 and 70)
* income: median income of subject’s neighborhood (3 levels)

## What roles will these variables play?

a1c is our outcome, which we’ll predict using three models …

1. Model 1: Use a1c\_old alone to predict a1c
2. Model 2: Use a1c\_old and age together to predict a1c
3. Model 3: Use a1c\_old, age, and income together to predict a1c

## The dm500 data

dm500 <- readRDS("c19/data/dm500.Rds")  
  
dm500

# A tibble: 500 × 5  
 a1c a1c\_old age income subject  
 <dbl> <dbl> <dbl> <fct> <chr>   
 1 6.3 11.4 62 Higher\_than\_50K S-001   
 2 11 16.3 54 Between\_30-50K S-002   
 3 8.7 10.7 47 <NA> S-003   
 4 6.5 5.8 53 Below\_30K S-004   
 5 6.7 6.3 64 Between\_30-50K S-005   
 6 5.8 6.5 48 Below\_30K S-006   
 7 9.6 NA 49 Below\_30K S-007   
 8 6.1 7.2 63 Between\_30-50K S-008   
 9 12.9 7.7 55 Below\_30K S-009   
10 6.7 6.8 63 Below\_30K S-010   
# ℹ 490 more rows

## Single Imputation

Today, we’ll assume all missing values are Missing at Random (MAR) and create 10 imputations but just use the 7th.

set.seed(20241031)  
  
dm500\_tenimps <- mice(dm500, m = 10, printFlag = FALSE)

Warning: Number of logged events: 1

dm500\_i <- complete(dm500\_tenimps, 7) |> tibble()  
  
n\_miss(dm500)

[1] 24

n\_miss(dm500\_i)

[1] 0

* Later, we’ll return and use all 10 imputations.

## Partitioning the Data

* Select a random sample (without replacement) of 70% of dm500\_i (60-80% is common) for model training.
* Hold out the other 30% for model testing, using anti\_join() to pull subjects not in dm500\_i\_train.

set.seed(4312024)  
  
dm500\_i\_train <- dm500\_i |>   
 slice\_sample(prop = 0.7, replace = FALSE)  
dm500\_i\_test <-   
 anti\_join(dm500\_i, dm500\_i\_train, by = "subject")  
  
c(nrow(dm500\_i\_train), nrow(dm500\_i\_test), nrow(dm500\_i))

[1] 350 150 500

## Three Regression Models We’ll Fit

* We continue to use the model training sample, and work with the (100/a1c) transformation.

fit1 <- lm((100/a1c) ~ a1c\_old, data = dm500\_i\_train)  
  
fit2 <- lm((100/a1c) ~ a1c\_old + age, data = dm500\_i\_train)  
  
fit3 <- lm((100/a1c) ~ a1c\_old + age + income,   
 data = dm500\_i\_train)  
  
c(n\_obs(fit1), n\_obs(fit2), n\_obs(fit3))

[1] 350 350 350

* n\_obs() gives us the number of observations used to fit the model. It comes from the insight package in easystats.

# Assess fit of candidate models in training sample.

## Estimated coefficients (fit1)

model\_parameters(fit1, ci = 0.95)

Parameter | Coefficient | SE | 95% CI | t(348) | p  
-------------------------------------------------------------------  
(Intercept) | 20.97 | 0.54 | [19.90, 22.04] | 38.49 | < .001  
a1c old | -0.98 | 0.07 | [-1.12, -0.85] | -14.34 | < .001

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

* Code: $$\hat{\frac{100}{A1c}} = 20.97 - 0.98 \mbox{ A1c\_old}$$

## Interpreting the fit1 equation (1/6)

* Interpret the Intercept?
* Our fit1 model estimates the value of (100/A1c) to be 20.97 if A1c\_old = 0, but we have no values of A1c\_old anywhere near 0 in our data[[1]](#footnote-31), nor is such a value plausible clinically, so the intercept doesn’t tell us much here.

## Interpreting the fit1 equation (2/6)

Our fit1 model estimates the slope of A1c\_old to be -0.98, with 95% CI (-1.12, -0.85). Interpret the point estimate…

* Suppose Harry has an A1c\_old value that is one percentage point (A1c is measured in percentage points) higher than Sally’s. Our fit1 model predicts the 100/A1c value for Harry to be 0.98 less than that of Sally, on average.

## Interpreting the fit1 equation (3/6)

Our fit1 model estimates the slope of A1c\_old to be -0.98, with 95% CI (-1.12, -0.85). What can we say about the CI?

* The confidence interval reflects imprecision in the population estimate, based only on assuming that the participants are selected at random from the population of interest.

## Interpreting the fit1 equation (4/6)

Slope of A1c\_old in fit1 is -0.98, with 95% CI (-1.12, -0.85).

* Model fit1 estimates a slope of -0.98 in study participants.
* When we generalize beyond study participants to the population they were selected at random from, then our data are **compatible** (at the 95% confidence level) with population slopes between -1.12 and -0.85, depending on the assumptions of our linear model fit1 being correct.

## Interpreting the fit1 equation (5/6)

* Practically, is our data a random sample of anything?

Slope of A1c\_old in fit1 is -0.98, with 95% CI (-1.12, -0.85).

* Our 95% confidence interval suggests that our data appear compatible with population slope values for A1c\_old between -1.12 and -0.85, assuming the participants are **representative** of the population of interest, **and** assuming the underlying linear model fit1 is correct.

## Interpreting the fit1 equation (6/6)

* Can we say “There is 95% probability that the population slope lies between …”?

To find such a probability interval, we’d need to **combine** our confidence interval (giving compatibility of data with population slope values) with **meaningful prior information**[[2]](#footnote-37) on which values for the population mean are plausible.

* For more, see [this article by Hilary Watt](https://academic.oup.com/ije/article/49/6/2083/5876177) 2020 Int J Epidemiology (<https://doi.org/10.1093/ije/dyaa080>)

## Summarize Fit Quality (fit1)

model\_performance(fit1)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
------------------------------------------------------------------  
1583.106 | 1583.176 | 1594.680 | 0.371 | 0.370 | 2.303 | 2.309

* Adjusted = , where = number of predictors in the model, and = number of observations.
  + Adjusted is no longer a percentage of anything, just an index. Higher values, though, still indicate stronger fits.
* RMSE/Sigma = Residual Standard Error -> smaller values = better fit

## Summarize Fit Quality (fit1)

model\_performance(fit1)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
------------------------------------------------------------------  
1583.106 | 1583.176 | 1594.680 | 0.371 | 0.370 | 2.303 | 2.309

* AIC = Akaike vs. BIC = Bayesian Information Criterion -> also smaller values = better fit
  + As we’ll see, the compare\_performance() function weights AIC and BIC measures so that higher values indicate a better fit.
  + Without the weights, we look for lower AIC and BIC.

## Estimated coefficients (fit2)

model\_parameters(fit2)

Parameter | Coefficient | SE | 95% CI | t(347) | p  
-------------------------------------------------------------------  
(Intercept) | 19.42 | 1.02 | [17.42, 21.43] | 19.06 | < .001  
a1c old | -0.96 | 0.07 | [-1.10, -0.82] | -13.79 | < .001  
age | 0.02 | 0.01 | [ 0.00, 0.05] | 1.79 | 0.074

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

* Suppose Harry is one year older than Sally and they have the same A1c\_old. On average, our fit2 model predicts Harry’s 100/A1c value to be 0.02 higher than Sally’s.

## Summarize Fit Quality (fit2)

model\_performance(fit1)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
------------------------------------------------------------------  
1583.106 | 1583.176 | 1594.680 | 0.371 | 0.370 | 2.303 | 2.309

model\_performance(fit2)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
------------------------------------------------------------------  
1581.884 | 1582.000 | 1597.316 | 0.377 | 0.374 | 2.292 | 2.302

* fit2 has higher , higher adjusted , lower RMSE, lower Sigma, lower values of AIC and corrected AIC.
* fit1 has a lower value of BIC.

## Compare Fit Quality (fit1 vs. fit2)

Remember compare\_performance() weights AIC and BIC so higher values indicate better fit.

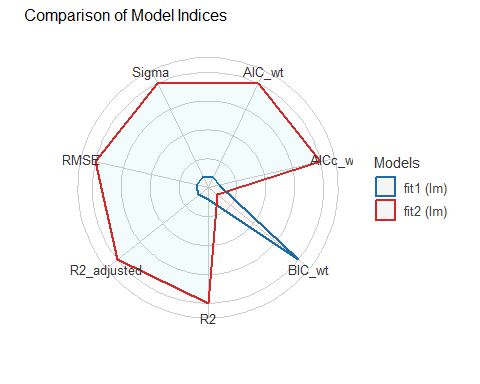
compare\_performance(fit1, fit2, rank = TRUE)

# Comparison of Model Performance Indices  
  
Name | Model | R2 | R2 (adj.) | RMSE | Sigma | AIC weights | AICc weights | BIC weights | Performance-Score  
---------------------------------------------------------------------------------------------------------------  
fit2 | lm | 0.377 | 0.374 | 2.292 | 2.302 | 0.648 | 0.643 | 0.211 | 85.71%  
fit1 | lm | 0.371 | 0.370 | 2.303 | 2.309 | 0.352 | 0.357 | 0.789 | 14.29%

* fit2 has higher , higher adjusted , lower RMSE, lower Sigma, higher (weighted) AIC and (weighted) AIC corrected
* fit1 has higher (weighted) BIC.

## fit1 vs. fit2 performance

plot(compare\_performance(fit1, fit2))



## Estimated coefficients (fit3)

model\_parameters(fit3)

Parameter | Coefficient | SE | 95% CI | t(345) | p  
-------------------------------------------------------------------------------  
(Intercept) | 19.49 | 1.07 | [17.39, 21.59] | 18.24 | < .001  
a1c old | -0.95 | 0.07 | [-1.09, -0.81] | -13.61 | < .001  
age | 0.02 | 0.01 | [ 0.00, 0.05] | 1.69 | 0.092   
income [Between\_30-50K] | 0.05 | 0.32 | [-0.58, 0.69] | 0.16 | 0.873   
income [Below\_30K] | -0.21 | 0.33 | [-0.87, 0.44] | -0.64 | 0.522

Uncertainty intervals (equal-tailed) and p-values (two-tailed) computed  
 using a Wald t-distribution approximation.

$$
\hat{\frac{100}{A1c}} = 19.49 - 0.95 \mbox{ A1c\_old} + 0.02 \mbox{ Age} \\ + 0.05 \mbox{(Inc 30-50)} - 0.21 \mbox{(Inc<30)}
$$

## Interpreting the slopes

$$
\hat{\frac{100}{A1c}} = 19.49 - 0.95 \mbox{ A1c\_old} + 0.02 \mbox{ Age} \\ + 0.05 \mbox{(Inc 30-50)} - 0.21 \mbox{(Inc<30)}
$$

* Suppose Harry and Sally are the same age and have the same A1c\_old, but Harry lives in a neighborhood with income < $30K, while Sally lives in a neighborhood with income > $50K. On average, our fit3 model predicts Harry’s 100/A1c value to be 0.21 lower than Sally’s.

## Summarize Fit Quality (All 3 models)

model\_performance(fit1)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
------------------------------------------------------------------  
1583.106 | 1583.176 | 1594.680 | 0.371 | 0.370 | 2.303 | 2.309

model\_performance(fit2)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
------------------------------------------------------------------  
1581.884 | 1582.000 | 1597.316 | 0.377 | 0.374 | 2.292 | 2.302

model\_performance(fit3)

# Indices of model performance  
  
AIC | AICc | BIC | R2 | R2 (adj.) | RMSE | Sigma  
------------------------------------------------------------------  
1584.938 | 1585.183 | 1608.086 | 0.379 | 0.372 | 2.289 | 2.306

## Compare Fit Quality (3 models)

Remember compare\_performance() weights AIC and BIC so higher values indicate better fit.

compare\_performance(fit1, fit2, fit3, rank = TRUE)

# Comparison of Model Performance Indices  
  
Name | Model | R2 | R2 (adj.) | RMSE | Sigma | AIC weights | AICc weights | BIC weights | Performance-Score  
---------------------------------------------------------------------------------------------------------------  
fit2 | lm | 0.377 | 0.374 | 2.292 | 2.302 | 0.568 | 0.568 | 0.211 | 83.06%  
fit3 | lm | 0.379 | 0.372 | 2.289 | 2.306 | 0.123 | 0.116 | 9.67e-04 | 43.26%  
fit1 | lm | 0.371 | 0.370 | 2.303 | 2.309 | 0.308 | 0.316 | 0.788 | 26.54%

## Which Model Looks Best? (1/4)

| Model |  | Adjusted | Predictors |
| --- | --- | --- | --- |
| fit1 | 0.371 | 0.370 | a1c\_old |
| fit2 | 0.377 | 0.374 | a1c\_old, age |
| fit3 | 0.379 | 0.372 | a1c\_old, age, income |

* By , the largest model (fit3) always looks best (raw is greedy)
* Adjusted penalizes for lack of parsimony. fit2 looks best.

## More Performance Indices (2/4)

| Model | RMSE | Sigma | Predictors |
| --- | --- | --- | --- |
| fit1 | 2.303 | 2.309 | a1c\_old |
| fit2 | 2.292 | 2.302 | a1c\_old, age |
| fit3 | 2.289 | 2.306 | a1c\_old, age, income |

* For and RMSE, smaller values, indicate better fits.
  + fit3 looks best by RMSE.
  + fit2 looks best by Sigma ().

## Still More Performance Indices (3/4)

Unweighted versions, from model\_performance()…

| Model | AIC | AIC\_c | BIC | Predictors |
| --- | --- | --- | --- | --- |
| fit1 | 1583.106 | 1583.176 | 1594.680 | a1c\_old |
| fit2 | 1581.884 | 1582.000 | 1597.316 | + age |
| fit3 | 1584.938 | 1585.183 | 1608.086 | + income |

* For unweighted AIC (both types) and BIC, smaller values (more negative, if relevant) indicate better fits.
  + fit2 looks best by AIC and corrected AIC.
  + fit1 looks best by BIC.

## Weighted Performance Indices (4/4)

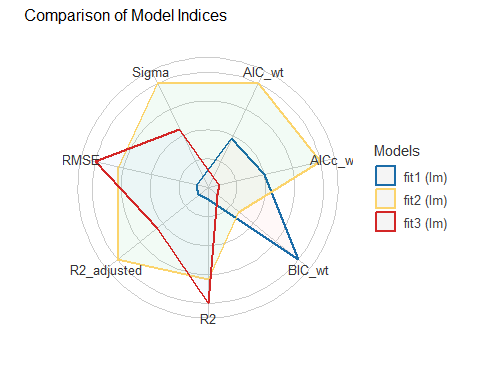
**WEIGHTED** versions, from compare\_performance()…

| Model | AIC (wtd) | AIC\_c (wtd) | BIC (wtd) | Predictors |
| --- | --- | --- | --- | --- |
| fit1 | 0.308 | 0.316 | 0.788 | a1c\_old |
| fit2 | 0.568 | 0.568 | 0.211 | + age |
| fit3 | 0.123 | 0.116 | 9.7e-04 | + income |

* After weighting of AIC (both types) and BIC, larger values indicate better fits.
  + fit2 looks best by AIC and corrected AIC.
  + fit1 looks best by BIC.

## Performance Indices for 3 Models

plot(compare\_performance(fit1, fit2, fit3))



# Check regression assumptions in training sample.

## Assessing the models with check\_model()

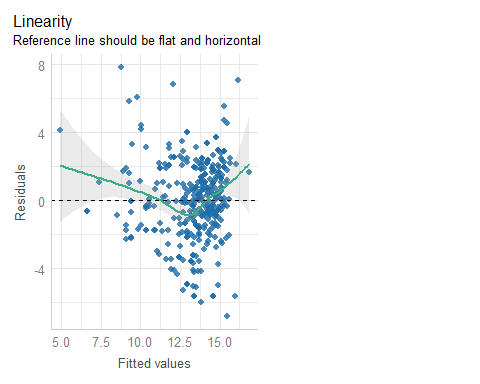
Three key assumptions we need to think about:

1. Linearity
2. Constant Variance (Homoscedasticity)
3. Normality

How do we assess 1, 2, and 3? Residual plots.

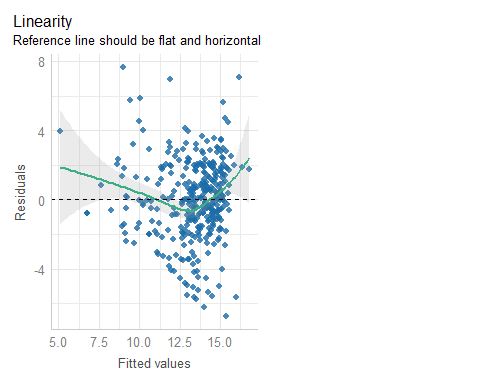
## Checking Linearity for fit1

check\_model(fit1, check = "linearity")



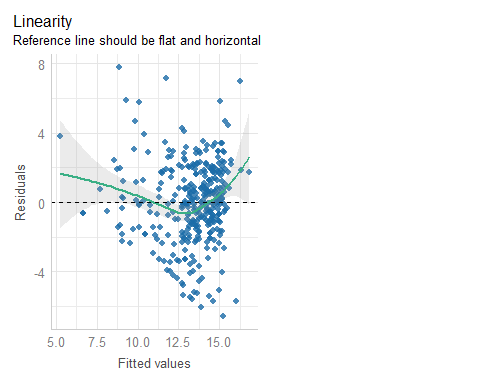
## Checking Linearity for fit2

check\_model(fit2, check = "linearity")



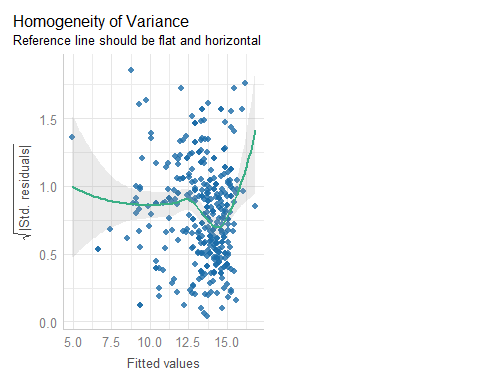
## Checking Linearity for fit3

check\_model(fit3, check = "linearity")



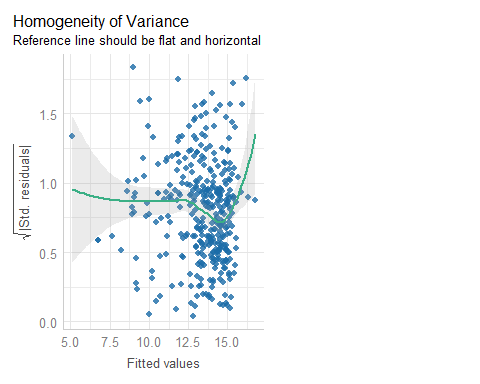
## Checking Constant Variance for fit1

check\_model(fit1, check = "homogeneity")



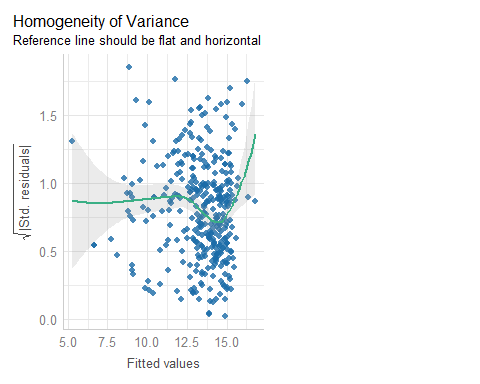
## Checking Constant Variance for fit2

check\_model(fit2, check = "homogeneity")



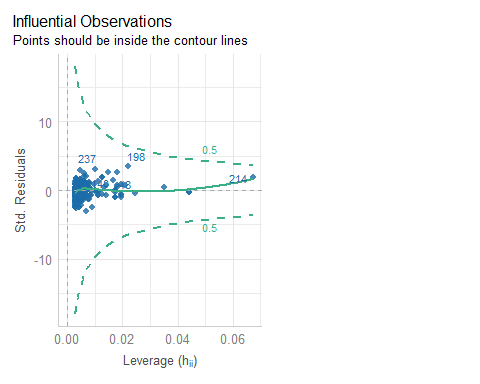
## Checking Constant Variance for fit3

check\_model(fit3, check = "homogeneity")



## Influential Observations in fit1?

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



## Looking at row 214?

dm500\_i\_train |> slice(214)

# A tibble: 1 × 5  
 a1c a1c\_old age income subject  
 <dbl> <dbl> <dbl> <fct> <chr>   
1 11 16.3 54 Between\_30-50K S-002

## augment adds fits, residuals, etc.

from the broom package…

aug1 <- augment(fit1, data = dm500\_i\_train) |>  
 mutate(inv\_a1c = 100/a1c) # add in our model's outcome

aug1 includes all variables in dm500\_i\_train and also:

* inv\_a1c = 100/a1c, transformed outcome fit1 predicts
* .fitted = fitted (predicted) values of 1/a1c
* .resid = residual (observed - fitted outcome) values; larger residuals (positive or negative) mean poorer fit
* .std.resid = standardized residuals (residuals scaled to SD = 1, remember residual mean is already 0)

## What does augment give us?

aug1 also includes:

* .hat statistic = measures *leverage* (larger values of .hat indicate unusual combinations of predictor values)
* .cooksd = Cook’s distance (or Cook’s d), a measure of the subject’s *influence* on the model (larger Cook’s d values indicate that removing the point will materially change the model’s coefficients)
* plus .sigma = estimated if this point is dropped from the model

## augment results: last 6 subjects

aug1 |> tail()

# A tibble: 6 × 12  
 a1c a1c\_old age income subject .fitted .resid .hat .sigma .cooksd  
 <dbl> <dbl> <dbl> <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 5.9 6.2 64 Higher\_than… S-035 14.9 2.07 0.00496 2.31 2.01e-3  
2 11 8.7 60 Below\_30K S-458 12.4 -3.33 0.00365 2.31 3.84e-3  
3 6.4 7.3 67 Below\_30K S-436 13.8 1.83 0.00303 2.31 9.53e-4  
4 7.8 7.7 49 Below\_30K S-363 13.4 -0.587 0.00286 2.31 9.28e-5  
5 6.6 5.6 56 Higher\_than… S-058 15.5 -0.318 0.00691 2.31 6.63e-5  
6 10.2 8.6 37 Between\_30-… S-102 12.5 -2.72 0.00350 2.31 2.44e-3  
# ℹ 2 more variables: .std.resid <dbl>, inv\_a1c <dbl>

aug1 |> tail() |> select(subject, .std.resid, inv\_a1c)

# A tibble: 6 × 3  
 subject .std.resid inv\_a1c  
 <chr> <dbl> <dbl>  
1 S-035 0.898 16.9   
2 S-458 -1.45 9.09  
3 S-436 0.792 15.6   
4 S-363 -0.254 12.8   
5 S-058 -0.138 15.2   
6 S-102 -1.18 9.80

## Summarizing our New Measures

for model fit1…

aug1 |> select(inv\_a1c, .fitted:.std.resid) |> summary()

inv\_a1c .fitted .resid .hat   
 Min. : 5.988 Min. : 4.963 Min. :-6.84847 Min. :0.002859   
 1st Qu.:11.528 1st Qu.:12.720 1st Qu.:-1.34925 1st Qu.:0.003037   
 Median :13.699 Median :13.800 Median : 0.04099 Median :0.003831   
 Mean :13.360 Mean :13.360 Mean : 0.00000 Mean :0.005714   
 3rd Qu.:15.385 3rd Qu.:14.585 3rd Qu.: 1.57236 3rd Qu.:0.005543   
 Max. :23.256 Max. :16.844 Max. : 7.87436 Max. :0.067183   
 .sigma .cooksd .std.resid   
 Min. :2.273 Min. :0.0000000 Min. :-2.975936   
 1st Qu.:2.309 1st Qu.:0.0001615 1st Qu.:-0.585759   
 Median :2.311 Median :0.0009335 Median : 0.017804   
 Mean :2.309 Mean :0.0034349 Mean : 0.000472   
 3rd Qu.:2.312 3rd Qu.:0.0029273 3rd Qu.: 0.685871   
 Max. :2.313 Max. :0.1330338 Max. : 3.447828

## augment results: row 214

for model fit1…

aug1 |> slice(214)

# A tibble: 1 × 12  
 a1c a1c\_old age income subject .fitted .resid .hat .sigma .cooksd  
 <dbl> <dbl> <dbl> <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 11 16.3 54 Between\_30-5… S-002 4.96 4.13 0.0672 2.30 0.123  
# ℹ 2 more variables: .std.resid <dbl>, inv\_a1c <dbl>

aug1 |> slice(214) |> select(subject, .std.resid, inv\_a1c)

# A tibble: 1 × 3  
 subject .std.resid inv\_a1c  
 <chr> <dbl> <dbl>  
1 S-002 1.85 9.09

## augment results: row 198

for model fit1…

aug1 |> slice(198)

# A tibble: 1 × 12  
 a1c a1c\_old age income subject .fitted .resid .hat .sigma .cooksd  
 <dbl> <dbl> <dbl> <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 6 12.4 59 Below\_30K S-105 8.79 7.87 0.0219 2.27 0.133  
# ℹ 2 more variables: .std.resid <dbl>, inv\_a1c <dbl>

aug1 |> slice(198) |> select(subject, .std.resid, inv\_a1c)

# A tibble: 1 × 3  
 subject .std.resid inv\_a1c  
 <chr> <dbl> <dbl>  
1 S-105 3.45 16.7

## Testing the largest outlier?

outlierTest(fit1) ## from car package

No Studentized residuals with Bonferroni p < 0.05  
Largest |rstudent|:  
 rstudent unadjusted p-value Bonferroni p  
198 3.503225 0.00051981 0.18193

A studentized residual is just another way to standardize the residuals that has some useful properties here.

* No indication that having a maximum absolute value of 3.5 in a sample of 350 studentized residuals is a major concern about the Normality assumption, given the Bonferroni p-value = 0.182.

## augment for models fit2 and fit3

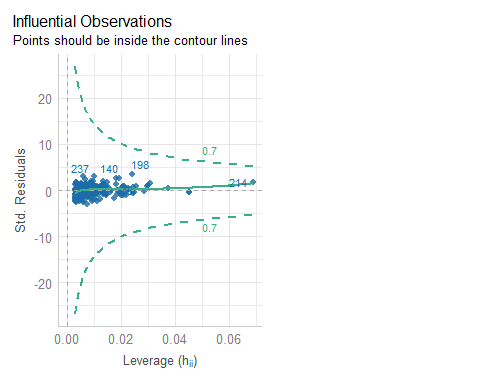
Later, we’ll need the augment results for our other two models: fit2 and fit3.

aug2 <- augment(fit2, data = dm500\_i\_train) |>  
 mutate(inv\_a1c = 100/a1c) # add in our model's outcome

aug3 <- augment(fit3, data = dm500\_i\_train) |>  
 mutate(inv\_a1c = 100/a1c) # add in our model's outcome

## Influential Observations in fit2?

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



## Summary across fit2

aug2 |> select(inv\_a1c, .fitted:.std.resid) |> summary()

inv\_a1c .fitted .resid .hat   
 Min. : 5.988 Min. : 5.128 Min. :-6.7622 Min. :0.002868   
 1st Qu.:11.528 1st Qu.:12.705 1st Qu.:-1.2724 1st Qu.:0.004416   
 Median :13.699 Median :13.765 Median : 0.0109 Median :0.006527   
 Mean :13.360 Mean :13.360 Mean : 0.0000 Mean :0.008571   
 3rd Qu.:15.385 3rd Qu.:14.593 3rd Qu.: 1.6531 3rd Qu.:0.010009   
 Max. :23.256 Max. :16.774 Max. : 7.6779 Max. :0.068792   
 .sigma .cooksd .std.resid   
 Min. :2.267 Min. :0.0000000 Min. :-2.948425   
 1st Qu.:2.302 1st Qu.:0.0001642 1st Qu.:-0.557132   
 Median :2.304 Median :0.0009053 Median : 0.004754   
 Mean :2.302 Mean :0.0030790 Mean : 0.000599   
 3rd Qu.:2.305 3rd Qu.:0.0028486 3rd Qu.: 0.720881   
 Max. :2.305 Max. :0.0940908 Max. : 3.376360

## augment results: row 214

for model fit2…

aug2 |> slice(214)

# A tibble: 1 × 12  
 a1c a1c\_old age income subject .fitted .resid .hat .sigma .cooksd  
 <dbl> <dbl> <dbl> <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 11 16.3 54 Between\_30-5… S-002 5.13 3.96 0.0688 2.29 0.0784  
# ℹ 2 more variables: .std.resid <dbl>, inv\_a1c <dbl>

aug2 |> slice(214) |> select(subject, .std.resid, inv\_a1c)

# A tibble: 1 × 3  
 subject .std.resid inv\_a1c  
 <chr> <dbl> <dbl>  
1 S-002 1.78 9.09

## augment results: row 198

for model fit2…

aug2 |> slice(198)

# A tibble: 1 × 12  
 a1c a1c\_old age income subject .fitted .resid .hat .sigma .cooksd  
 <dbl> <dbl> <dbl> <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 6 12.4 59 Below\_30K S-105 8.99 7.68 0.0242 2.27 0.0941  
# ℹ 2 more variables: .std.resid <dbl>, inv\_a1c <dbl>

aug2 |> slice(198) |> select(subject, .std.resid, inv\_a1c)

# A tibble: 1 × 3  
 subject .std.resid inv\_a1c  
 <chr> <dbl> <dbl>  
1 S-105 3.38 16.7

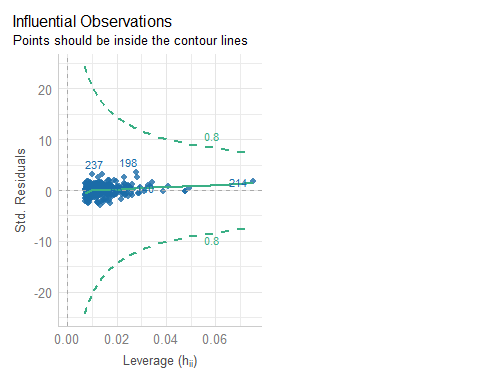
## Testing the largest outlier?

outlierTest(fit2) ## from car package

No Studentized residuals with Bonferroni p < 0.05  
Largest |rstudent|:  
 rstudent unadjusted p-value Bonferroni p  
198 3.428276 0.0006807 0.23824

## Influential Observations in fit3?

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



## Summary across fit3

aug3 |> select(inv\_a1c, .fitted:.std.resid) |> summary()

inv\_a1c .fitted .resid .hat   
 Min. : 5.988 Min. : 5.29 Min. :-6.593697 Min. :0.007099   
 1st Qu.:11.528 1st Qu.:12.75 1st Qu.:-1.347919 1st Qu.:0.009490   
 Median :13.699 Median :13.74 Median : 0.003505 Median :0.012570   
 Mean :13.360 Mean :13.36 Mean : 0.000000 Mean :0.014286   
 3rd Qu.:15.385 3rd Qu.:14.62 3rd Qu.: 1.674379 3rd Qu.:0.016285   
 Max. :23.256 Max. :16.81 Max. : 7.810286 Max. :0.075026   
 .sigma .cooksd .std.resid   
 Min. :2.269 Min. :0.0000000 Min. :-2.879152   
 1st Qu.:2.305 1st Qu.:0.0001981 1st Qu.:-0.587438   
 Median :2.308 Median :0.0011510 Median : 0.001538   
 Mean :2.306 Mean :0.0029771 Mean : 0.000606   
 3rd Qu.:2.309 3rd Qu.:0.0030504 3rd Qu.: 0.729660   
 Max. :2.309 Max. :0.0675136 Max. : 3.435729

## augment results: row 214

for model fit3…

aug3 |> slice(214)

# A tibble: 1 × 12  
 a1c a1c\_old age income subject .fitted .resid .hat .sigma .cooksd  
 <dbl> <dbl> <dbl> <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 11 16.3 54 Between\_30-5… S-002 5.29 3.80 0.0750 2.30 0.0477  
# ℹ 2 more variables: .std.resid <dbl>, inv\_a1c <dbl>

aug3|> slice(214) |> select(subject, .std.resid, inv\_a1c)

# A tibble: 1 × 3  
 subject .std.resid inv\_a1c  
 <chr> <dbl> <dbl>  
1 S-002 1.71 9.09

## augment results: row 198

for model fit3…

aug3 |> slice(198)

# A tibble: 1 × 12  
 a1c a1c\_old age income subject .fitted .resid .hat .sigma .cooksd  
 <dbl> <dbl> <dbl> <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 6 12.4 59 Below\_30K S-105 8.86 7.81 0.0278 2.27 0.0675  
# ℹ 2 more variables: .std.resid <dbl>, inv\_a1c <dbl>

aug3 |> slice(198) |> select(subject, .std.resid, inv\_a1c)

# A tibble: 1 × 3  
 subject .std.resid inv\_a1c  
 <chr> <dbl> <dbl>  
1 S-105 3.44 16.7

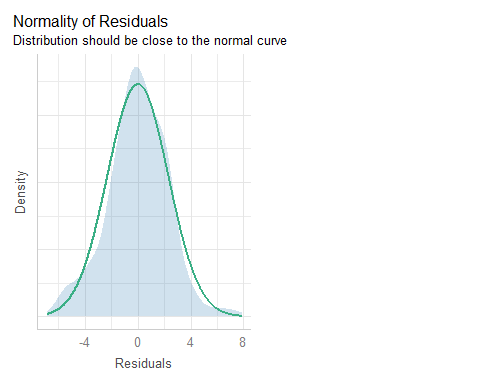
## Testing the largest outlier?

outlierTest(fit3) ## from car package

No Studentized residuals with Bonferroni p < 0.05  
Largest |rstudent|:  
 rstudent unadjusted p-value Bonferroni p  
198 3.490988 0.00054392 0.19037

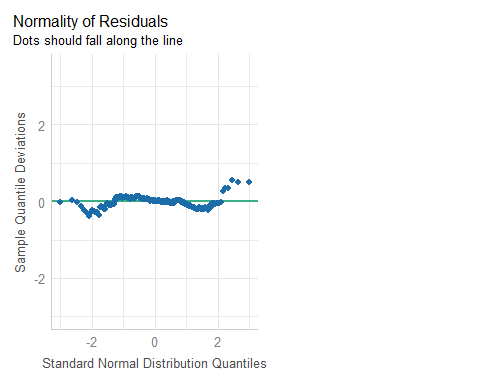
## Normality Check of fit1 Residuals?

check\_model(fit1, check = "normality")



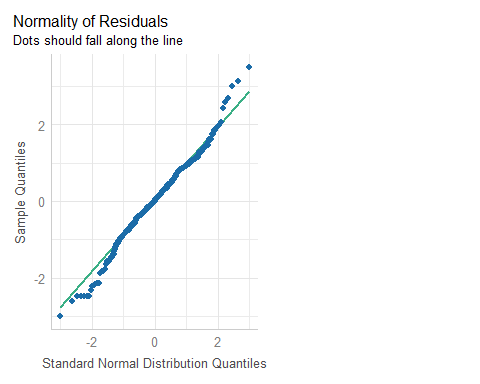
## Normal Q-Q with detrending?

check\_model(fit1, check = "qq", detrend = TRUE)



## Normal Q-Q without detrending?

check\_model(fit1, check = "qq", detrend = FALSE)



## Identifying poorly fit points

which.max(aug1$.std.resid)

[1] 198

which.min(aug1$.std.resid)

[1] 1

slice(aug1, c(1, 198))

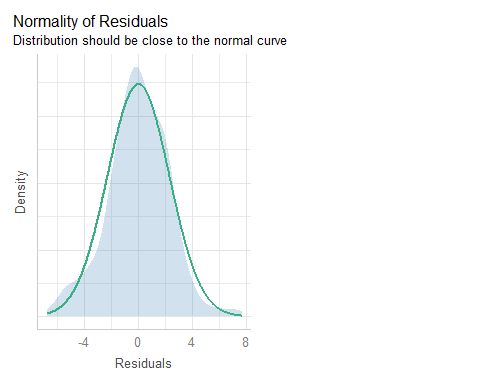
# A tibble: 2 × 12  
 a1c a1c\_old age income subject .fitted .resid .hat .sigma .cooksd  
 <dbl> <dbl> <dbl> <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 11.6 5.6 54 Below\_30K S-168 15.5 -6.85 0.00691 2.28 0.0308  
2 6 12.4 59 Below\_30K S-105 8.79 7.87 0.0219 2.27 0.133   
# ℹ 2 more variables: .std.resid <dbl>, inv\_a1c <dbl>

slice(aug1, c(1, 198)) |> select(subject, .std.resid, inv\_a1c)

# A tibble: 2 × 3  
 subject .std.resid inv\_a1c  
 <chr> <dbl> <dbl>  
1 S-168 -2.98 8.62  
2 S-105 3.45 16.7

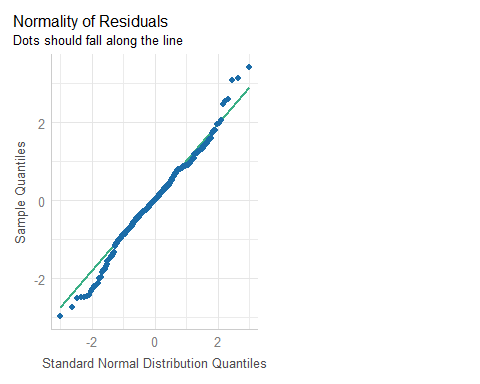
## Normality Check of fit2 Residuals?

check\_model(fit2, check = "normality")



## Normal Q-Q of Residuals in fit2?

check\_model(fit2, check = "qq", detrend = FALSE)



## Identifying poorly fit points

which.max(aug2$.std.resid)

[1] 198

which.min(aug2$.std.resid)

[1] 1

slice(aug2, c(1, 198))

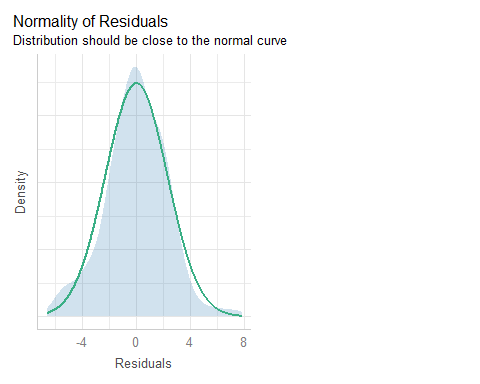
# A tibble: 2 × 12  
 a1c a1c\_old age income subject .fitted .resid .hat .sigma .cooksd  
 <dbl> <dbl> <dbl> <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 11.6 5.6 54 Below\_30K S-168 15.4 -6.76 0.00735 2.28 0.0215  
2 6 12.4 59 Below\_30K S-105 8.99 7.68 0.0242 2.27 0.0941  
# ℹ 2 more variables: .std.resid <dbl>, inv\_a1c <dbl>

slice(aug2, c(1, 198)) |> select(subject, .std.resid, inv\_a1c)

# A tibble: 2 × 3  
 subject .std.resid inv\_a1c  
 <chr> <dbl> <dbl>  
1 S-168 -2.95 8.62  
2 S-105 3.38 16.7

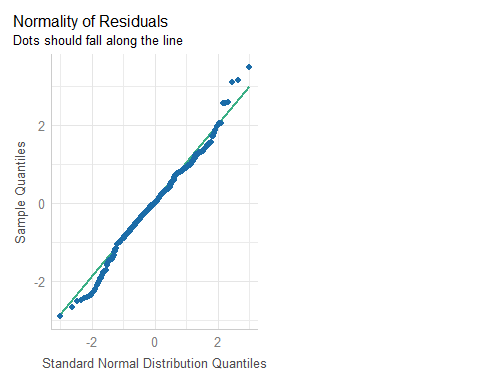
## Normality Check of fit3 Residuals?

check\_model(fit3, check = "normality")



## Normal Q-Q of Residuals in fit3?

check\_model(fit3, check = "qq", detrend = FALSE)



## Identifying poorly fit points

which.max(aug3$.std.resid)

[1] 198

which.min(aug3$.std.resid)

[1] 1

slice(aug3, c(1, 198))

# A tibble: 2 × 12  
 a1c a1c\_old age income subject .fitted .resid .hat .sigma .cooksd  
 <dbl> <dbl> <dbl> <fct> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 11.6 5.6 54 Below\_30K S-168 15.2 -6.59 0.0133 2.28 0.0223  
2 6 12.4 59 Below\_30K S-105 8.86 7.81 0.0278 2.27 0.0675  
# ℹ 2 more variables: .std.resid <dbl>, inv\_a1c <dbl>

slice(aug3, c(1, 198)) |> select(subject, .std.resid, inv\_a1c)

# A tibble: 2 × 3  
 subject .std.resid inv\_a1c  
 <chr> <dbl> <dbl>  
1 S-168 -2.88 8.62  
2 S-105 3.44 16.7

## Checking for Collinearity

Collinearity refers to correlations between predictors.

* Strong correlations between predictors can inflate the variance of our estimates, and also make it difficult to distinguish effects attributable to the correlated predictors.
* When we have multiple predictors (as in fit2 and fit3) it is worthwhile to quantify the extent to which our variances are inflated by this collinearity.

Sometimes, a correlation matrix or a scatterplot matrix can help…

## Correlation Matrix (from Class 19)

temp <- dm500\_i\_train |> mutate(inv\_a1c = 100/a1c) |>   
 select(a1c\_old, age, income, inv\_a1c)  
correlation(temp)

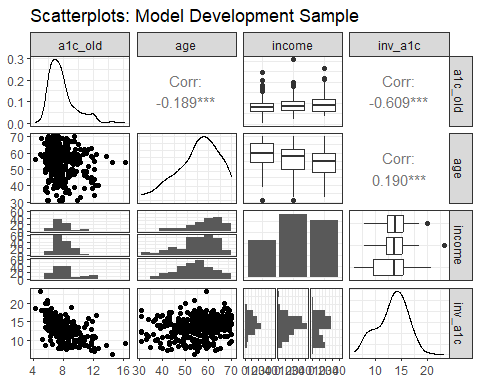
# Correlation Matrix (pearson-method)  
  
Parameter1 | Parameter2 | r | 95% CI | t(348) | p  
---------------------------------------------------------------------  
a1c\_old | age | -0.19 | [-0.29, -0.09] | -3.59 | < .001\*\*\*  
a1c\_old | inv\_a1c | -0.61 | [-0.67, -0.54] | -14.34 | < .001\*\*\*  
age | inv\_a1c | 0.19 | [ 0.09, 0.29] | 3.60 | < .001\*\*\*  
  
p-value adjustment method: Holm (1979)  
Observations: 350

* fit2 includes age and a1c\_old as predictors
* fit3 adds in a categorical predictor: income category (not shown)

## Scatterplot Matrix (from Class 19)

* I select the outcome last. Then, the bottom row will show the most important scatterplots, with the outcome on the Y axis, and each predictor, in turn on the X.
* ggpairs() comes from the GGally package.

temp <- dm500\_i\_train |>   
 mutate(inv\_a1c = 100/a1c) |>  
 select(a1c\_old, age, income, inv\_a1c)  
  
ggpairs(temp,   
 title = "Scatterplots: Model Development Sample",  
 lower = list(combo = wrap("facethist", bins = 10)))



## The vif() function

We can estimate the effect of collinearity directly through the vif() function in the car package.

vif(fit3)

GVIF Df GVIF^(1/(2\*Df))  
a1c\_old 1.047487 1 1.023468  
age 1.066866 1 1.032892  
income 1.045901 2 1.011283

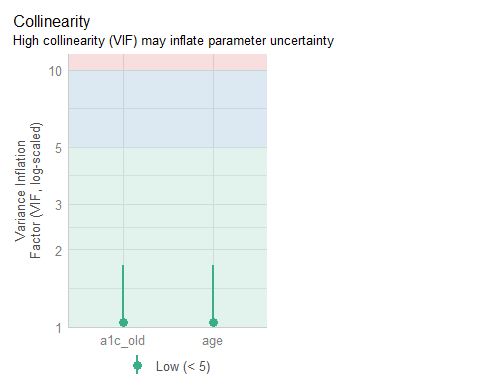
vif(fit2)

a1c\_old age   
1.036986 1.036986

* (generalized) variance inflation factors above 5 are worthy of our special attention

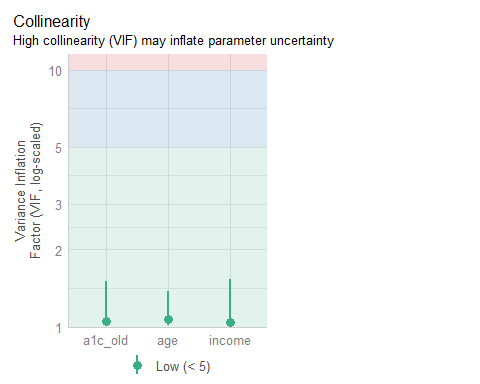
## Model fit2 Check for Collinearity

check\_model(fit2, check = "vif")



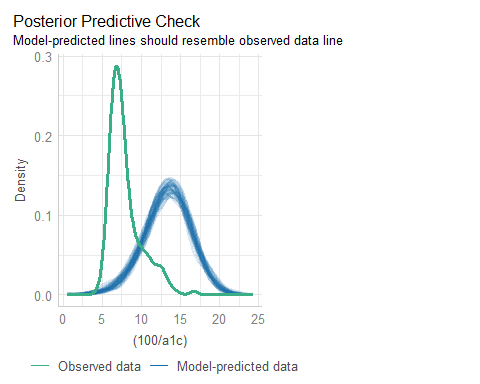
## Model fit3 Check for Collinearity

check\_model(fit3, check = "vif")



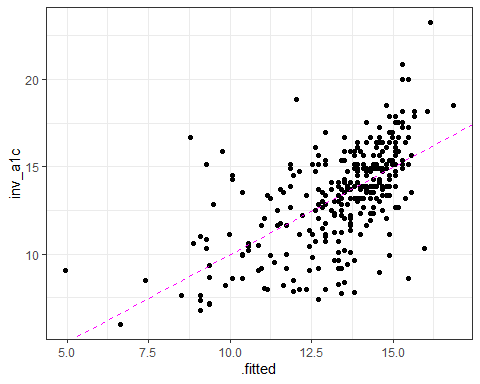
## Posterior Predictive Check: fit1

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



## fit1: Observed vs. Predicted

ggplot(aug1, aes(x = .fitted, y = inv\_a1c)) +  
 geom\_point() +  
 geom\_abline(slope = 1, intercept = 0, lty = "dashed", col = "magenta")



## fit1: Observed and Predicted

aug1 |> reframe(lovedist(inv\_a1c))

# A tibble: 1 × 10  
 n miss mean sd med mad min q25 q75 max  
 <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 350 0 13.4 2.91 13.7 2.86 5.99 11.5 15.4 23.3

aug1 |> reframe(lovedist(.fitted))

# A tibble: 1 × 10  
 n miss mean sd med mad min q25 q75 max  
 <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 350 0 13.4 1.77 13.8 1.31 4.96 12.7 14.6 16.8

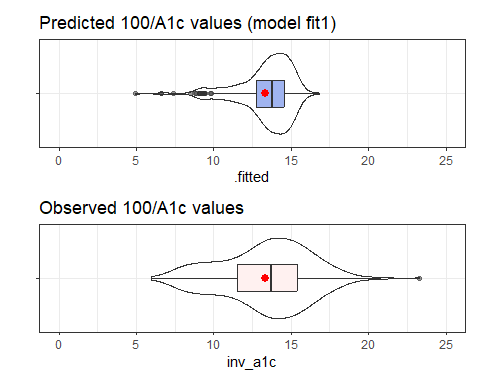
cor(aug1$inv\_a1c, aug1$.fitted)

[1] 0.609385

## fit1: Predictions and Observed

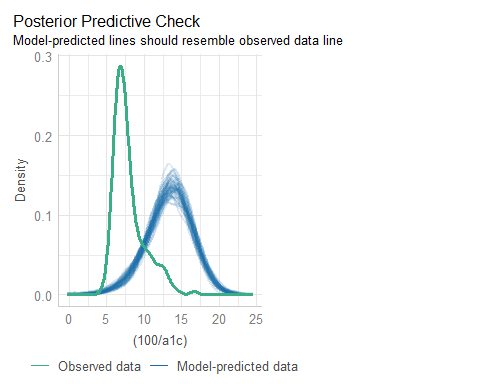
p1 <- ggplot(aug1, aes(x = .fitted, y = "")) +  
 geom\_violin() +   
 geom\_boxplot(fill = "royalblue", alpha = 0.5, width = 0.3) +   
 xlim(0, 25) +  
 stat\_summary(fun = "mean", col = "red") +  
 labs(title = "Predicted 100/A1c values (model fit1)", y = "")  
  
p2 <- ggplot(aug1, aes(x = inv\_a1c, y = "")) +  
 geom\_violin() +   
 geom\_boxplot(fill = "mistyrose", alpha = 0.5, width = 0.3) +   
 xlim(0, 25) +  
 stat\_summary(fun = "mean", col = "red") +  
 labs(title = "Observed 100/A1c values", y = "")  
  
p1 / p2

Warning: Removed 1 row containing missing values or values outside the scale range  
(`geom\_segment()`).  
Removed 1 row containing missing values or values outside the scale range  
(`geom\_segment()`).



## Posterior Predictive Check: fit2

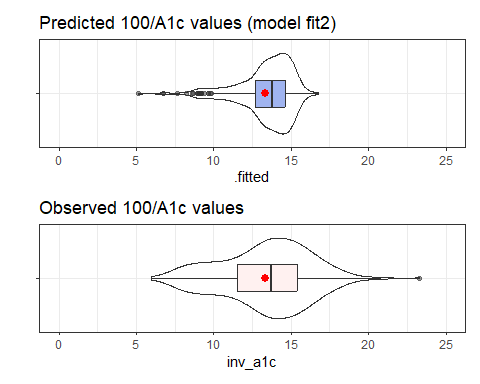
check\_model(fit2, check = "pp\_check")



## fit2: Predictions and Observed

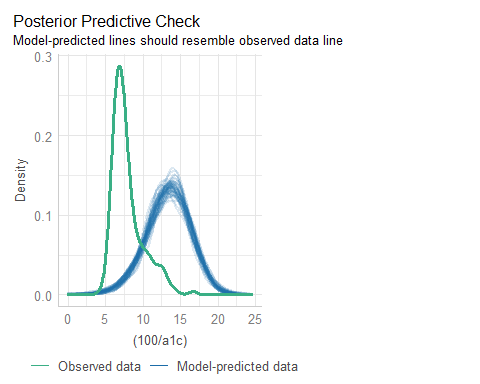
p1 <- ggplot(aug2, aes(x = .fitted, y = "")) +  
 geom\_violin() +   
 geom\_boxplot(fill = "royalblue", alpha = 0.5, width = 0.3) +   
 xlim(0, 25) +  
 stat\_summary(fun = "mean", col = "red") +  
 labs(title = "Predicted 100/A1c values (model fit2)", y = "")  
  
p2 <- ggplot(aug2, aes(x = inv\_a1c, y = "")) +  
 geom\_violin() +   
 geom\_boxplot(fill = "mistyrose", alpha = 0.5, width = 0.3) +   
 xlim(0, 25) +  
 stat\_summary(fun = "mean", col = "red") +  
 labs(title = "Observed 100/A1c values", y = "")  
  
p1 / p2

Warning: Removed 1 row containing missing values or values outside the scale range  
(`geom\_segment()`).  
Removed 1 row containing missing values or values outside the scale range  
(`geom\_segment()`).



## Posterior Predictive Check: fit3

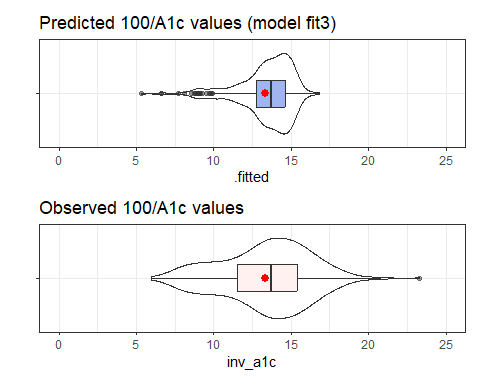
check\_model(fit3, check = "pp\_check")



## fit3: Predictions and Observed

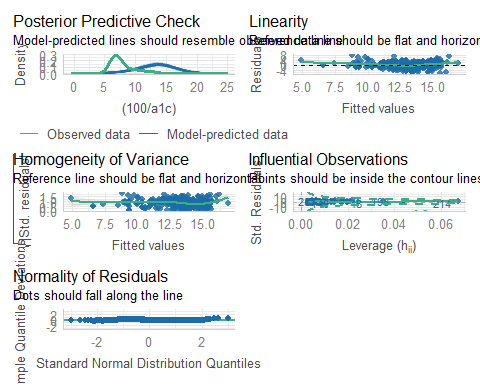
p1 <- ggplot(aug3, aes(x = .fitted, y = "")) +  
 geom\_violin() +   
 geom\_boxplot(fill = "royalblue", alpha = 0.5, width = 0.3) +   
 xlim(0, 25) +  
 stat\_summary(fun = "mean", col = "red") +  
 labs(title = "Predicted 100/A1c values (model fit3)", y = "")  
  
p2 <- ggplot(aug1, aes(x = inv\_a1c, y = "")) +  
 geom\_violin() +   
 geom\_boxplot(fill = "mistyrose", alpha = 0.5, width = 0.3) +   
 xlim(0, 25) +  
 stat\_summary(fun = "mean", col = "red") +  
 labs(title = "Observed 100/A1c values", y = "")  
  
p1 / p2

Warning: Removed 1 row containing missing values or values outside the scale range  
(`geom\_segment()`).  
Removed 1 row containing missing values or values outside the scale range  
(`geom\_segment()`).



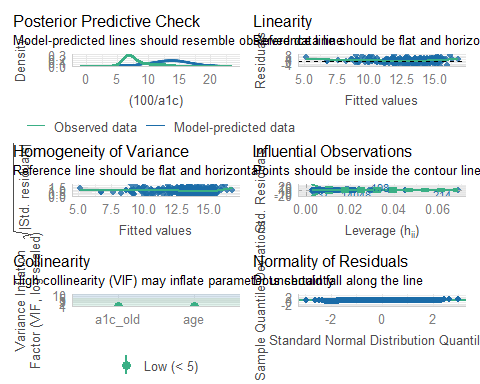
## Model fit1 Checking

check\_model(fit1)



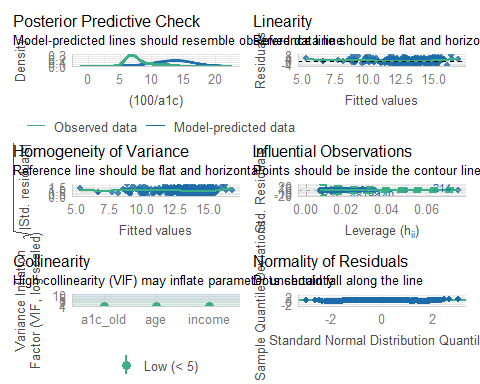
## Model fit2 Checking

check\_model(fit2)



## Model fit3 Checking

check\_model(fit3)



## Coming Soon

1. Assessing the candidate models more thoroughly, in both the training and test samples
   * MAPE, RMSPE, Maximum Prediction Error, Validated
2. Considering Bayesian alternative fits with weakly informative priors
3. Incorporating multiple imputation in building a final model

## Session Information

xfun::session\_info()

R version 4.4.1 (2024-06-14 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 askpass\_1.2.1 backports\_1.5.0   
 base64enc\_0.1.3 bayestestR\_0.15.0 bit\_4.5.0   
 bit64\_4.5.2 blob\_1.2.4 boot\_1.3-31   
 broom\_1.0.7 bslib\_0.8.0 cachem\_1.1.0   
 callr\_3.7.6 car\_3.1-3 carData\_3.0-5   
 cellranger\_1.1.0 cli\_3.6.3 clipr\_0.8.0   
 coda\_0.19-4.1 codetools\_0.2-20 colorspace\_2.1-1   
 compiler\_4.4.1 conflicted\_1.2.0 correlation\_0.8.6   
 cowplot\_1.1.3 cpp11\_0.5.0 crayon\_1.5.3   
 curl\_5.2.3 data.table\_1.16.2 datasets\_4.4.1   
 datawizard\_0.13.0 DBI\_1.2.3 dbplyr\_2.5.0   
 Deriv\_4.1.6 digest\_0.6.37 doBy\_4.6.24   
 dplyr\_1.1.4 dtplyr\_1.3.1 easystats\_0.7.3   
 effectsize\_0.8.9 emmeans\_1.10.5 estimability\_1.5.1   
 evaluate\_1.0.1 fansi\_1.0.6 farver\_2.1.2   
 fastmap\_1.2.0 fontawesome\_0.5.2 forcats\_1.0.0   
 foreach\_1.5.2 Formula\_1.2-5 fs\_1.6.4   
 gargle\_1.5.2 generics\_0.1.3 GGally\_2.2.1   
 ggplot2\_3.5.1 ggrepel\_0.9.6 ggstats\_0.7.0   
 glmnet\_4.1-8 glue\_1.8.0 googledrive\_2.1.1   
 googlesheets4\_1.1.1 graphics\_4.4.1 grDevices\_4.4.1   
 grid\_4.4.1 gridExtra\_2.3 gtable\_0.3.6   
 haven\_2.5.4 highr\_0.11 hms\_1.1.3   
 htmltools\_0.5.8.1 httr\_1.4.7 ids\_1.0.1   
 insight\_0.20.5 isoband\_0.2.7 iterators\_1.0.14   
 janitor\_2.2.0 jomo\_2.7-6 jquerylib\_0.1.4   
 jsonlite\_1.8.9 knitr\_1.48 labeling\_0.4.3   
 lattice\_0.22-6 lifecycle\_1.0.4 lme4\_1.1-35.5   
 lubridate\_1.9.3 magrittr\_2.0.3 MASS\_7.3-61   
 Matrix\_1.7-0 MatrixModels\_0.5.3 memoise\_2.0.1   
 methods\_4.4.1 mgcv\_1.9-1 mice\_3.16.0   
 microbenchmark\_1.5.0 mime\_0.12 minqa\_1.2.8   
 mitml\_0.4-5 modelbased\_0.8.9 modelr\_0.1.11   
 multcomp\_1.4-26 munsell\_0.5.1 mvtnorm\_1.3-1   
 naniar\_1.1.0 nlme\_3.1-164 nloptr\_2.1.1   
 nnet\_7.3-19 norm\_1.0.11.1 numDeriv\_2016.8.1.1   
 openssl\_2.2.2 ordinal\_2023.12.4.1 pan\_1.9   
 parallel\_4.4.1 parameters\_0.23.0 patchwork\_1.3.0   
 pbkrtest\_0.5.3 performance\_0.12.4 pillar\_1.9.0   
 pkgconfig\_2.0.3 plyr\_1.8.9 prettyunits\_1.2.0   
 processx\_3.8.4 progress\_1.2.3 ps\_1.8.0   
 purrr\_1.0.2 quantreg\_5.99 R6\_2.5.1   
 ragg\_1.3.3 rappdirs\_0.3.3 RColorBrewer\_1.1-3   
 Rcpp\_1.0.13 RcppEigen\_0.3.4.0.2 readr\_2.1.5   
 readxl\_1.4.3 rematch\_2.0.0 rematch2\_2.1.2   
 report\_0.5.9 reprex\_2.1.1 rlang\_1.1.4   
 rmarkdown\_2.28 rpart\_4.1.23 rstudioapi\_0.17.1   
 rvest\_1.0.4 sandwich\_3.1-1 sass\_0.4.9   
 scales\_1.3.0 see\_0.9.0 selectr\_0.4.2   
 shape\_1.4.6.1 snakecase\_0.11.1 SparseM\_1.84.2   
 splines\_4.4.1 stats\_4.4.1 stringi\_1.8.4   
 stringr\_1.5.1 survival\_3.7-0 sys\_3.4.3   
 systemfonts\_1.1.0 textshaping\_0.4.0 TH.data\_1.1-2   
 tibble\_3.2.1 tidyr\_1.3.1 tidyselect\_1.2.1   
 tidyverse\_2.0.0 timechange\_0.3.0 tinytex\_0.53   
 tools\_4.4.1 tzdb\_0.4.0 ucminf\_1.2.2   
 UpSetR\_1.4.0 utf8\_1.2.4 utils\_4.4.1   
 uuid\_1.2.1 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\_0.48 xml2\_1.3.6   
 xtable\_1.8-4 yaml\_2.3.10 zoo\_1.8-12

1. Range of A1c\_old was 4.2 to 16.3 percentage points. [↑](#footnote-ref-31)
2. perhaps via a Bayesian model with an informative (not just weakly informative) prior. [↑](#footnote-ref-37)