431 Class 20

Thomas E. Love, Ph.D.

2022-11-15

Today's Agenda

- Redo the regression analyses for `dm1` but now using single imputation.

Version 2022-10-31 21:25:29

Today’s Packages

```r
options(dplyr.summarise.inform = FALSE)

library(simputation) # for single impuation
library(car) # for boxCox
library(GGally) # for ggpairs
library(glue) # for adding R results to labels
library(ggrepel) # help with residual plots
library(equatiomatic) # help with equation extraction
library(broom) # for tidying model output
library(kableExtra) # formatting tables
library(janitor); library(naniar); library(patchwork)
library(tidyverse)

theme_set(theme_bw())
```

From Class 18

```r
dm1 <- readRDS("c20/data/dm1.Rds")

dm1_cc <- dm1 |> drop_na()

dm1_imp <- dm1 |>
  filter(complete.cases(a1c, subject)) |>
  impute_rlm(a1c_old ~ age) |>
  impute_cart(income ~ age + a1c_old)
```

Partition imputed data from `dm1_imp`

This time, we'll build an 80% development, 20% holdout partition of the `dm1_imp` data, and we'll also change our random seed, just for fun.

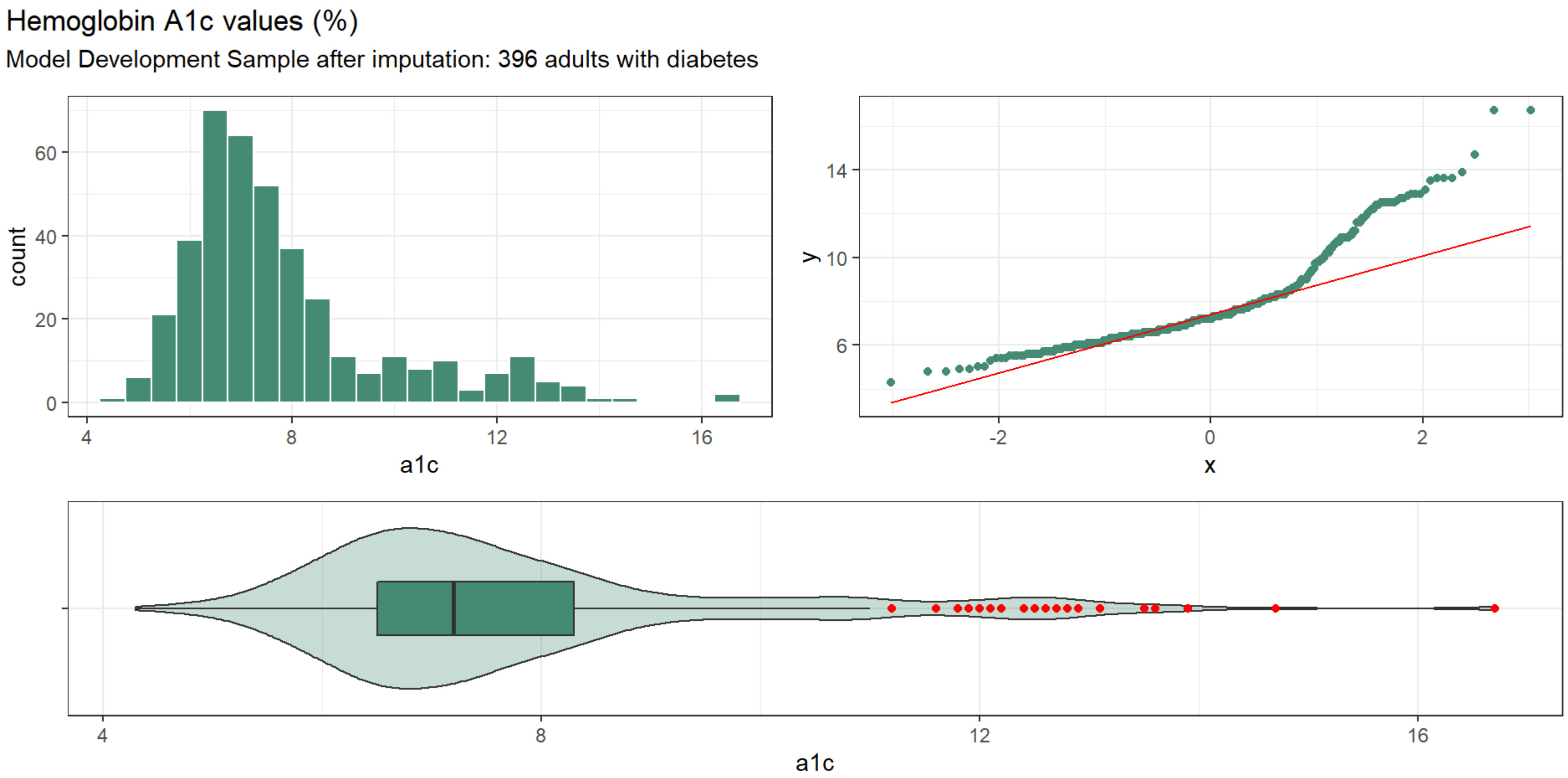
```r
set.seed(2022431)

dm1_imp_train <- dm1_imp |>
  slice_sample(prop = 0.8, replace = FALSE)

dm1_imp_test <-
  anti_join(dm1_imp, dm1_imp_train, by = "subject")

dim(dm1_imp_train); dim(dm1_imp_test)
```

```
[1] 396   5
[1] 100   5
```

Distribution of `a1c` in training sample

```r
p1 <- ggplot(dm1_imp_train, aes(x = a1c)) +
  geom_histogram(binwidth = 0.5,
                 fill = "aquamarine4", col = "white")

p2 <- ggplot(dm1_imp_train, aes(sample = a1c)) +
  geom_qq(col = "aquamarine4") + geom_qq_line(col = "red")

p3 <- ggplot(dm1_imp_train, aes(x = "", y = a1c)) +
  geom_violin(fill = "aquamarine4", alpha = 0.3) +
  geom_boxplot(fill = "aquamarine4", width = 0.3,
               outlier.color = "red") +
  labs(x = "") + coord_flip()

p1 + p2 - p3 +
  plot_layout(ncol = 1, height = c(3, 2)) +
  plot_annotation(title = "Hemoglobin A1c values (%)",
         subtitle = glue("Model Development Sample after imputation: ",
                           nrow(dm1_imp_train), " adults with diabetes"))
```

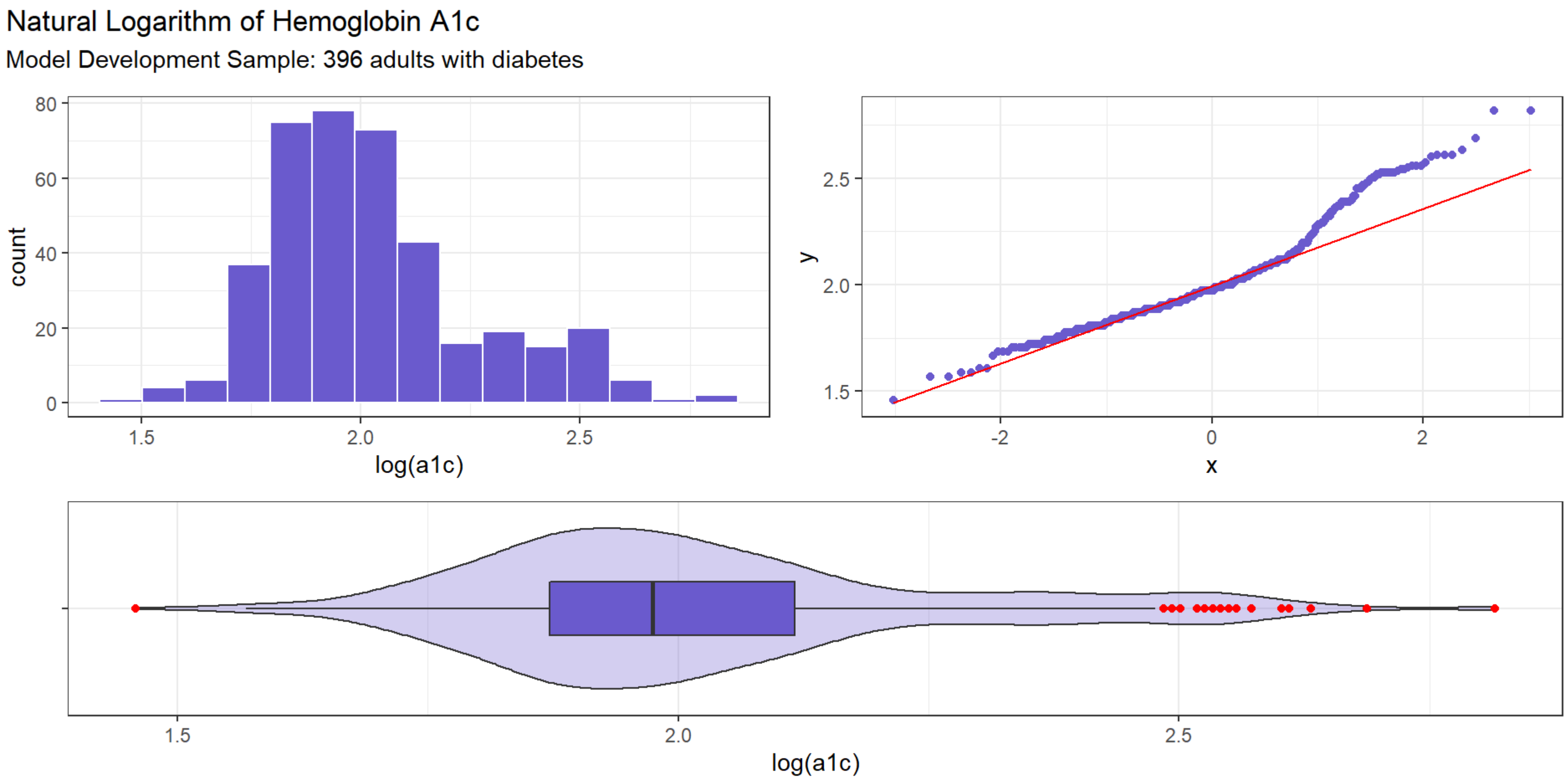
Distribution of a1c in training sample



Consider a log transformation?

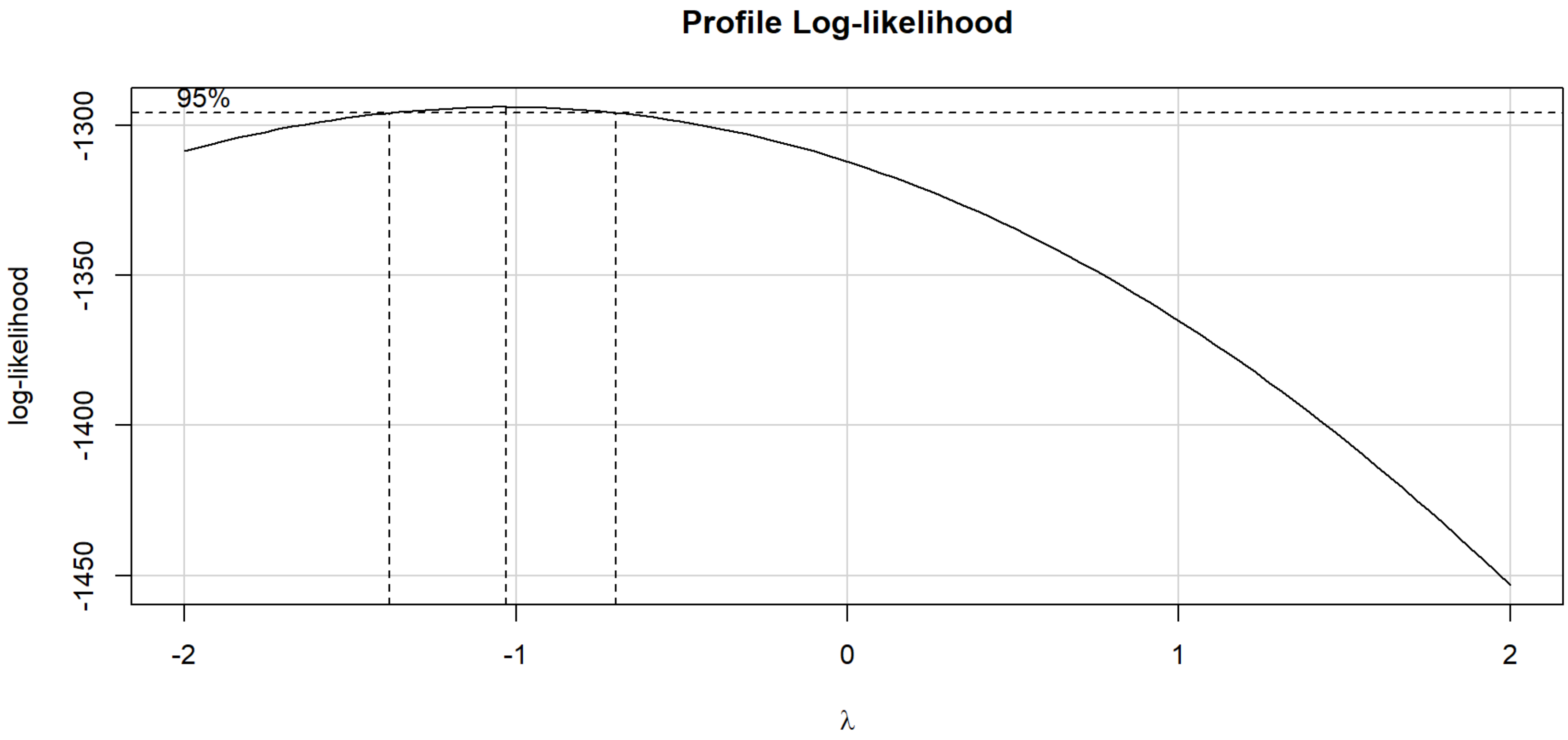
```r
p1 <- ggplot(dm1_imp_train, aes(x = log(a1c))) +
  geom_histogram(bins = 15,
                 fill = "slateblue", col = "white")

p2 <- ggplot(dm1_imp_train, aes(sample = log(a1c))) +
  geom_qq(col = "slateblue") + geom_qq_line(col = "red")

p3 <- ggplot(dm1_imp_train, aes(x = "", y = log(a1c))) +
  geom_violin(fill = "slateblue", alpha = 0.3) +
  geom_boxplot(fill = "slateblue", width = 0.3,
               outlier.color = "red") +
  labs(x = "") + coord_flip()

p1 + p2 - p3 +
  plot_layout(ncol = 1, height = c(3, 2)) +
  plot_annotation(title = "Natural Logarithm of Hemoglobin A1c",
         subtitle = paste0("Model Development Sample: ",
                           nrow(dm1_imp_train),
                           " adults with diabetes"))
```

Consider a log transformation?



What does Box-Cox suggest?

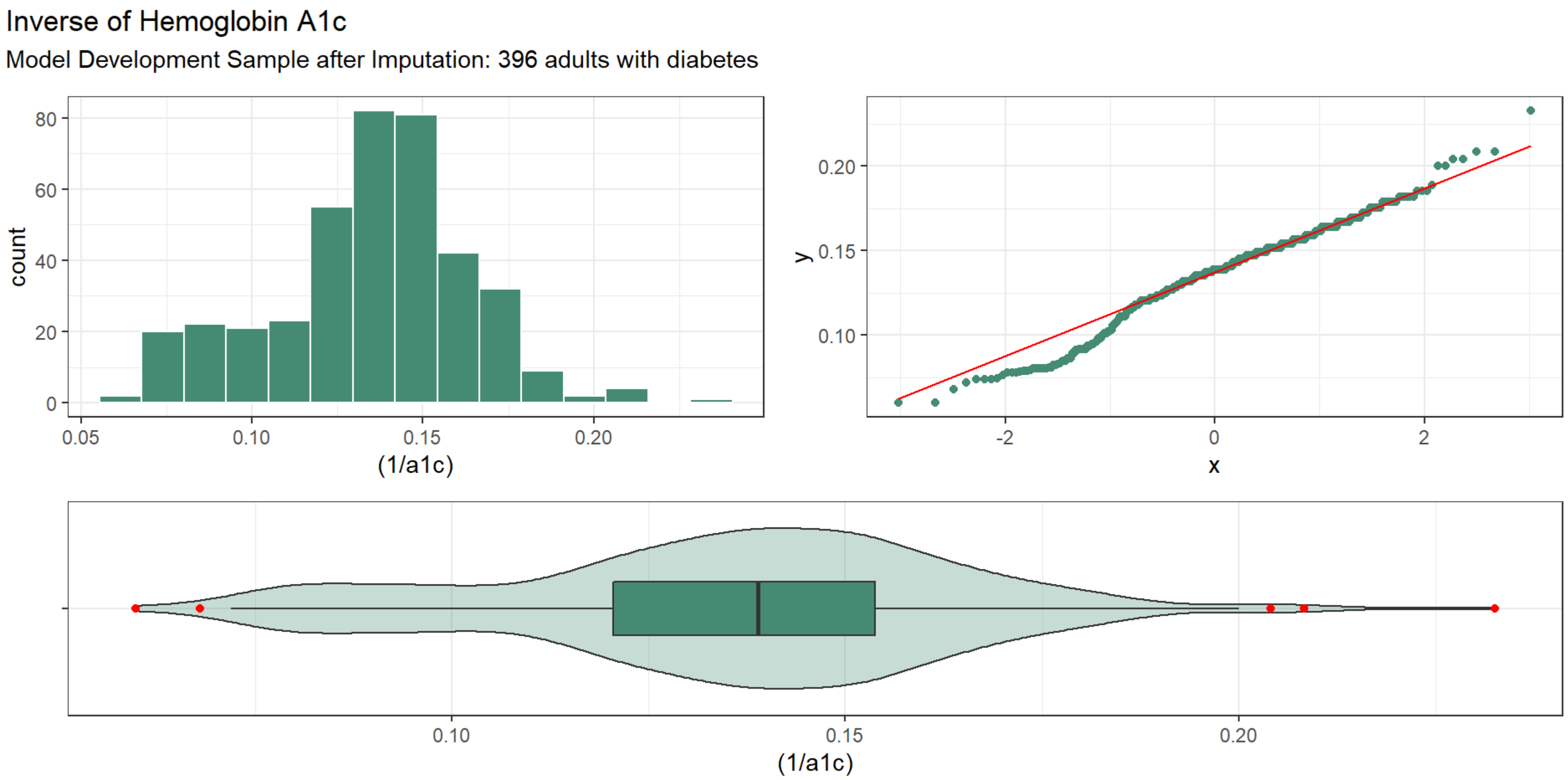
```
imod_0 <- lm(a1c ~ a1c_old + age + income,
             data = dm1_imp_train)
boxCox(imod_0)
```


Profile Log-likelihood

Inverse of A1c again?

```r
p1 <- ggplot(dm1_imp_train, aes(x = (1/a1c))) +
  geom_histogram(bins = 15,
                 fill = "aquamarine4", col = "white")

p2 <- ggplot(dm1_imp_train, aes(sample = (1/a1c))) +
  geom_qq(col = "aquamarine4") + geom_qq_line(col = "red")

p3 <- ggplot(dm1_imp_train, aes(x = "", y = (1/a1c))) +
  geom_violin(fill = "aquamarine4", alpha = 0.3) +
  geom_boxplot(fill = "aquamarine4", width = 0.3,
               outlier.color = "red") +
  labs(x = "") + coord_flip()

p1 + p2 - p3 +
  plot_layout(ncol = 1, height = c(3, 2)) +
  plot_annotation(title = "Inverse of Hemoglobin A1c",
         subtitle = paste0("Model Development Sample after Imputation: ",
                           nrow(dm1_imp_train),
                           " adults with diabetes"))
```

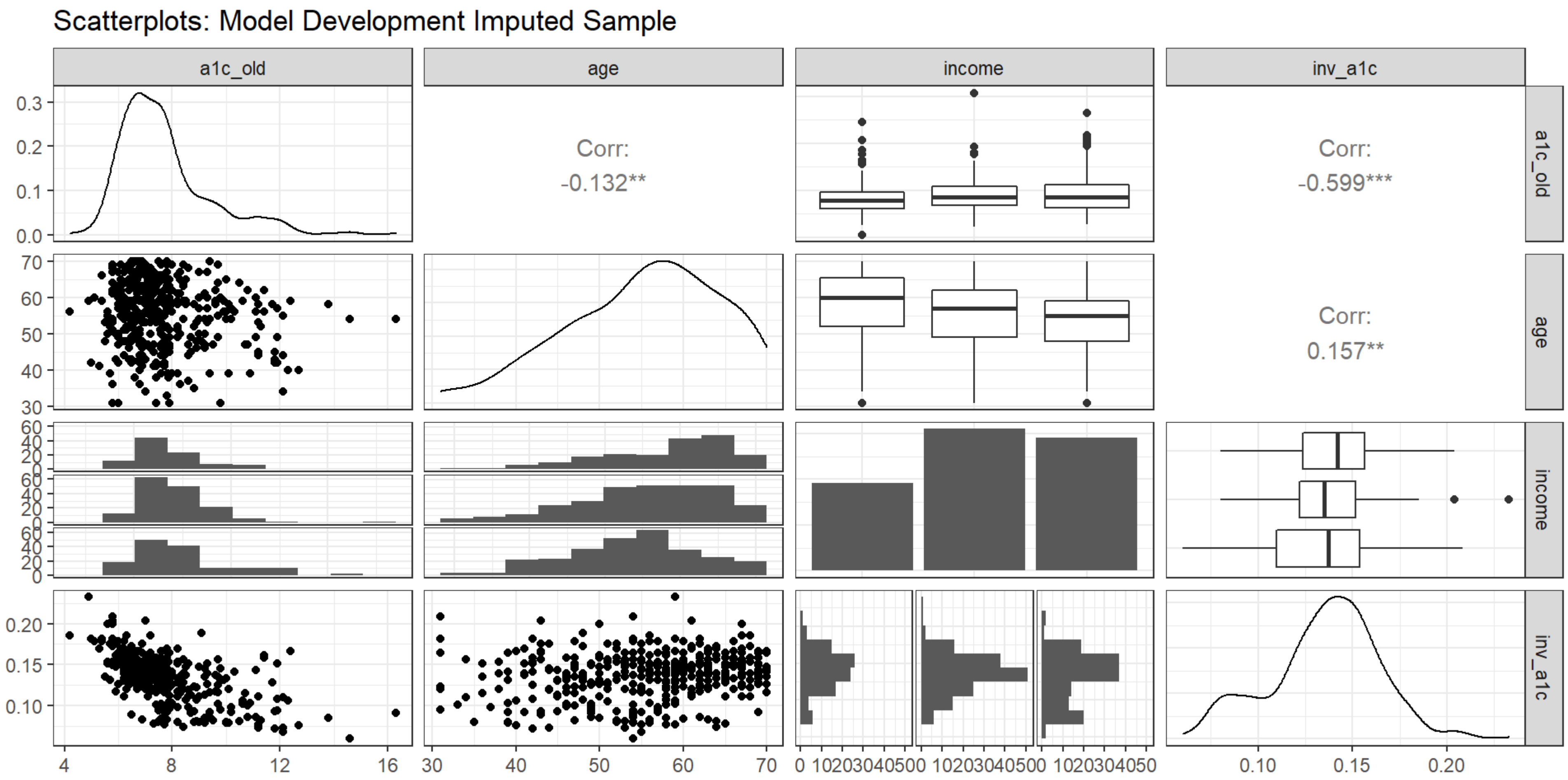
Inverse of A1c again?



Scatterplot Matrix

```r
temp <- dm1_imp_train |>
  mutate(inv_a1c = 1/a1c) |>
  select(a1c_old, age, income, inv_a1c)

ggpairs(temp,
    title = "Scatterplots: Model Development Imputed Sample",
    lower = list(combo = wrap("facethist", bins = 10)))
```

Scatterplot Matrix



Fitting the Same Three Models

- Remember we're using the model development sample here.

```
imod_1 <- lm((1/a1c) ~ a1c_old, data = dm1_imp_train)

imod_2 <- lm((1/a1c) ~ a1c_old + age, data = dm1_imp_train)

imod_3 <- lm((1/a1c) ~ a1c_old + age + income,
             data = dm1_imp_train)
```

Assess the quality of fit for candidate models within the development sample.

Tidied coefficients (imod_1)

```
tidy_im1 <- tidy(imod_1, conf.int = TRUE, conf.level = 0.95)

tidy_im1 |>
  select(term, estimate, std.error, p.value,
         conf.low, conf.high) |>
  kbl(digits = 4) |> kable_material(font_size = 28)
```

term	estimate	std.error	p.value	conf.low	conf.high
(Intercept)	0.2126	0.0053	0	0.2021	0.2231
a1c_old	-0.0101	0.0007	0	-0.0114	-0.0087

The Regression Equation (imod_1)

Again, we'll use the equatiomatic package.

```r
extract_eq(imod_1, use_coefs = TRUE, coef_digits = 4,
           ital_vars = TRUE, wrap = TRUE, terms_per_line = 3)
```

$$(\widehat{1/a1c}) = 0.2126 - 0.0101(a1c\_old)$$

Summary of Fit Quality (imod_1)

```r
glance(imod_1) |>
  mutate(name = "imod_1") |>
  select(name, r.squared, adj.r.squared,
         sigma, AIC, BIC) |>
  kbl(digits = c(0,3,3,3,0,0)) |> kable_minimal(font_size = 28)
```

name	r.squared	adj.r.squared	sigma	AIC	BIC
imod_1	0.359	0.357	0.023	-1857	-1845

Tidied coefficients (imod_2)

```r
tidy_im2 <- tidy(imod_2, conf.int = TRUE, conf.level = 0.95)

tidy_im2 |>
  select(term, estimate, std.error, p.value,
         conf.low, conf.high) |>
  kbl(digits = 4) |> kable_material(font_size = 28)
```

term	estimate	std.error	p.value	conf.low	conf.high
(Intercept)	0.1974	0.0094	0.0000	0.1789	0.2159
a1c_old	-0.0099	0.0007	0.0000	-0.0112	-0.0086
age	0.0002	0.0001	0.0507	0.0000	0.0005

The Regression Equation (imod_2)

Again, we'll use the equatiomatic package, and **results = 'asis'**.

```
extract_eq(imod_2, use_coefs = TRUE, coef_digits = 4,
           ital_vars = TRUE)
```

$$(\widehat{1/a1c}) = 0.1974 - 0.0099(a1c\_old) + 2e - 04(age)$$

Summary of Fit Quality (imod_2)

```r
glance(imod_2) |>
  mutate(name = "imod_2") |>
  select(name, r.squared, adj.r.squared,
         sigma, AIC, BIC) |>
  kbl(digits = c(0,3,3,3,0,0)) |> kable_minimal(font_size = 28)
```

name	r.squared	adj.r.squared	sigma	AIC	BIC
imod_2	0.365	0.362	0.023	-1859	-1843

Tidied coefficients (imod_3)

```r
tidy_im3 <- tidy(imod_3, conf.int = TRUE, conf.level = 0.95)

tidy_im3 |>
  select(term, estimate, se = std.error,
         low = conf.low, high = conf.high, p = p.value) |>
  kbl(digits = 4) |> kable_material(font_size = 28)
```

term	estimate	se	low	high	p
(Intercept)	0.1995	0.0098	0.1802	0.2188	0.0000
a1c_old	-0.0098	0.0007	-0.0112	-0.0085	0.0000
age	0.0002	0.0001	0.0000	0.0005	0.0749
incomeBetween_30-50K	-0.0013	0.0030	-0.0072	0.0047	0.6764
incomeBelow_30K	-0.0026	0.0031	-0.0087	0.0035	0.3966

The Regression Equation (`imod_3`)

Again, we'll use the `equatiomatic` package.

```r
extract_eq(imod_3, use_coefs = TRUE, coef_digits = 4,
           ital_vars = TRUE, wrap = TRUE, terms_per_line = 2)
```

$$
\begin{aligned}
\widehat{(1/a1c)} &= 0.1995 - 0.0098(a1c\_old) + \\
&\quad 2e-04(age) - 0.0013(income_{Between\_30-50} \\
&\quad 0.0026(income_{Below\_30K})
\end{aligned}
$$

Summary of Fit Quality (imod_3)

```r
glance(imod_3) |>
  mutate(name = "imod_3") |>
  select(name, r.squared, adj.r.squared,
         sigma, AIC, BIC) |>
  kbl(digits = c(0,3,3,3,0,0)) |> kable_minimal(font_size = 28)
```

name	r.squared	adj.r.squared	sigma	AIC	BIC
imod_3	0.366	0.36	0.023	-1855	-1832

I checked stepwise regression again

- Even though variable selection **never** works, it is seductive.

What if we do forward selection in this situation?

```
min.model <- lm(a1c ~ 1, data = dm1_imp_train)
fwd.model <- step(min.model, direction = "forward",
                  scope = ~ a1c_old + age + income)
```

```
Start:  AIC=564.64
a1c ~ 1

          Df Sum of Sq    RSS    AIC
+ a1c_old  1    627.64 1012.0 375.55
+ age      1     50.69 1588.9 554.20
+ income   2     38.51 1601.1 559.23
<none>                 1639.6 564.64

Step:  AIC=375.55
a1c ~ a1c_old
```

Stepwise Regression Results

We wind up back at the model with all three predictors in this case (`imod_3`).

```
fwd.model$coefficients
```

```
        (Intercept)               a1c_old                   age 
         3.25627910            0.71380675           -0.01893984 
incomeBetween_30-50K     incomeBelow_30K 
        -0.01834854            0.33534059 
```

- As we'll discuss in 432, there is an immense amount of evidence that variable selection causes severe problems in estimation and inference.

Which Model Looks Best In-Sample?

Considering each summary separately…

```
bind_rows(glance(imod_1), glance(imod_2), glance(imod_3)) |>
  mutate(model = c("imod_1", "imod_2", "imod_3"),
         vars = c("a1c_old", "+ age", "+ income")) |>
  select(model, vars, r2 = r.squared, adj_r2 = adj.r.squared,
         sigma, AIC, BIC) |>
  kbl(digits = c(0, 0, 3, 3, 5, 1, 0)) |> kable_classic(font_size = 28)
```

model	vars	r2	adj_r2	sigma	AIC	BIC
imod_1	a1c_old	0.359	0.357	0.02309	-1856.8	-1845
imod_2	+ age	0.365	0.362	0.02300	-1858.7	-1843
imod_3	+ income	0.366	0.360	0.02304	-1855.4	-1832

Conclusions from In-Sample Comparisons?

- `imod_3` (as it must, here) has the best R-square.
- `imod_2` wins on adjusted R-square and σ and AIC
- `imod_1` has the best BIC

Using augment to add fits, residuals, etc.

```
augi1 <- augment(imod_1, data = dm1_imp_train) |>
  mutate(inv_a1c = 1/a1c) # add in our model's outcome

augi2 <- augment(imod_2, data = dm1_imp_train) |>
  mutate(inv_a1c = 1/a1c) # add in our model's outcome

augi3 <- augment(imod_3, data = dm1_imp_train) |>
  mutate(inv_a1c = 1/a1c) # add in our model's outcome
```

Checking Regression Assumptions

Four key assumptions we need to think about:

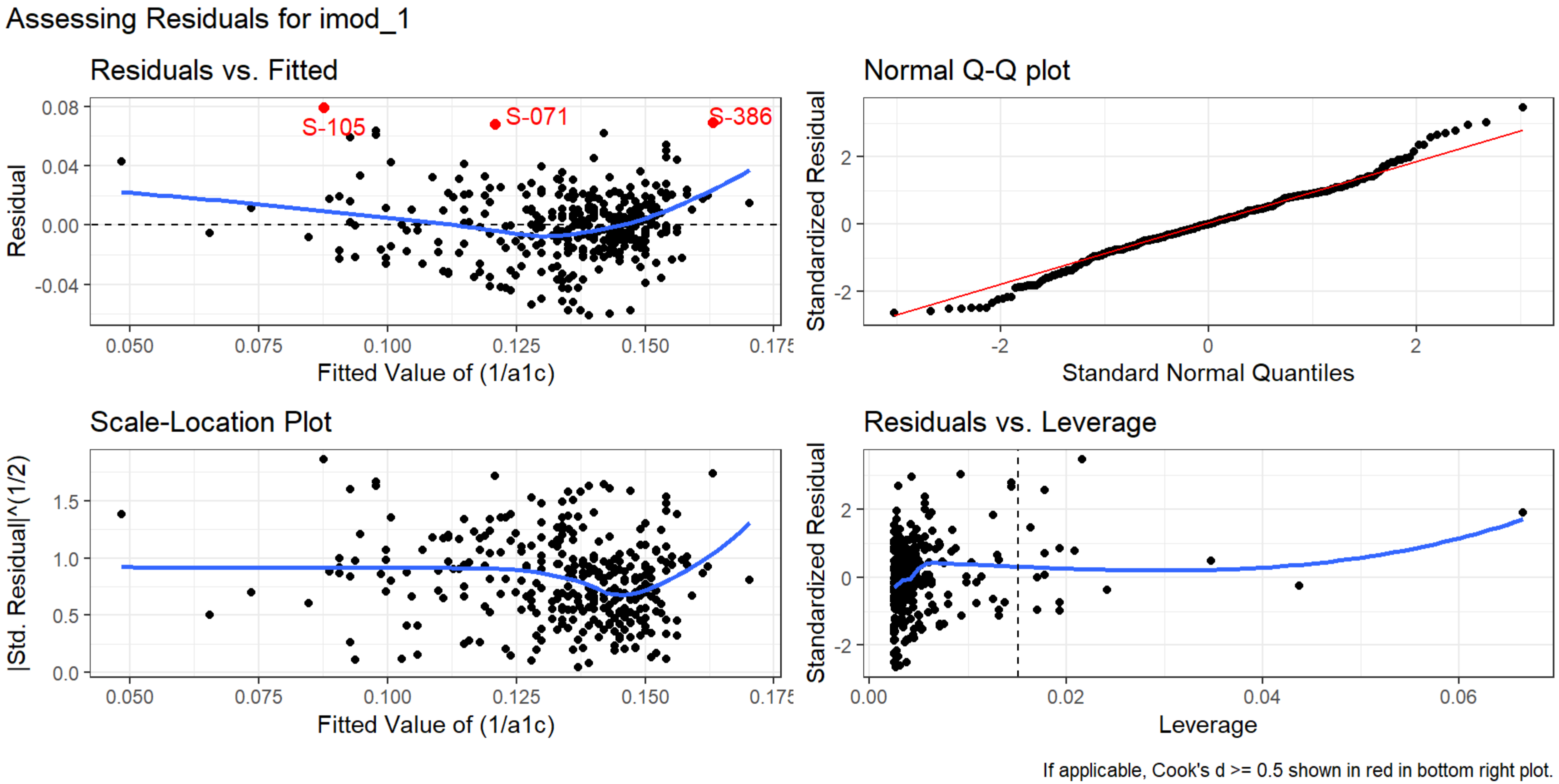
1. Linearity
2. Constant Variance (Homoscedasticity)
3. Normality
4. Independence

For each model, what can we say based on residual plots?

Residual Plots for imod_1 (via ggplot2)

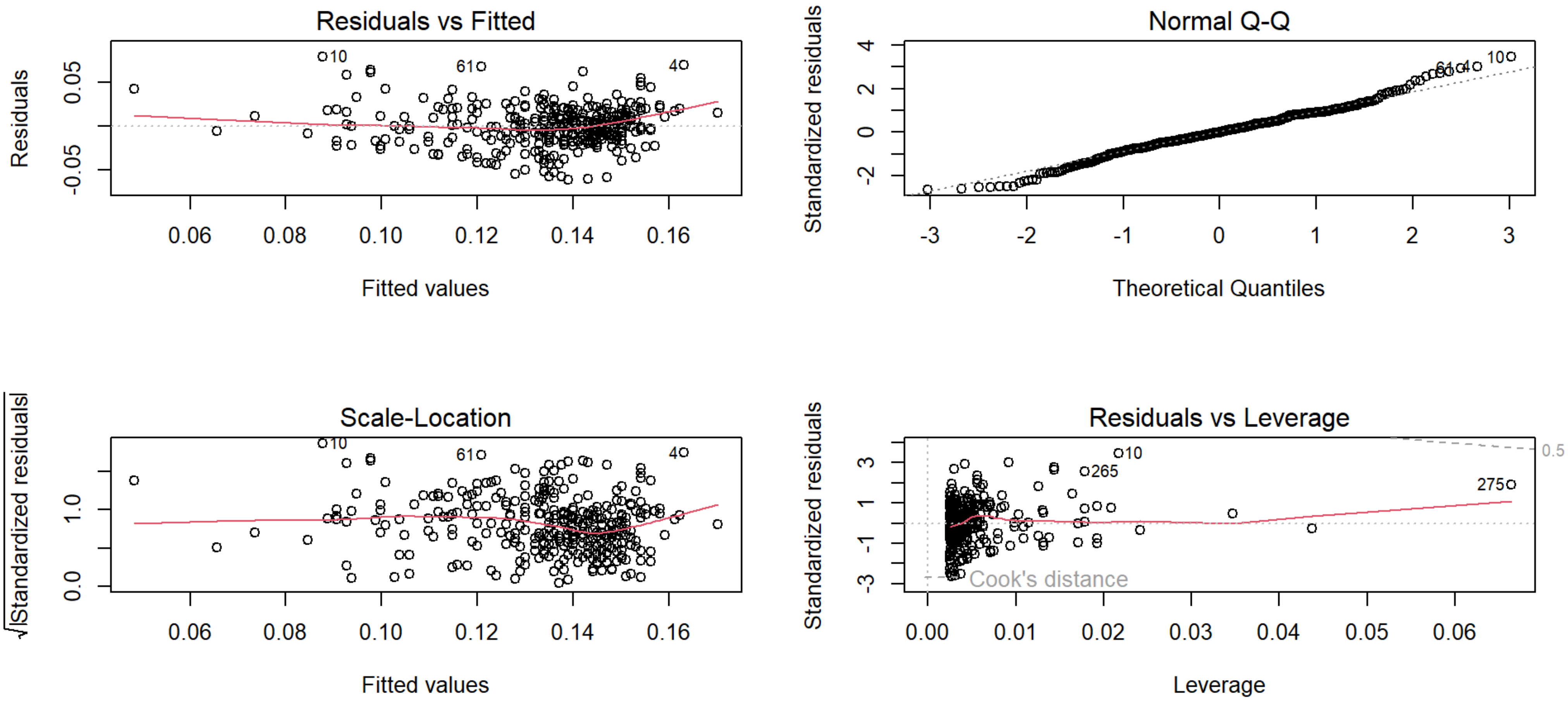
```r
p1 <- ggplot(augi1, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_point(data = augi1 |>
               slice_max(abs(.resid), n = 3),
             col = "red", size = 2) +
  geom_text_repel(data = augi1 |>
               slice_max(abs(.resid), n = 3),
               aes(label = subject), col = "red") +
  geom_abline(intercept = 0, slope = 0, lty = "dashed") +
  geom_smooth(method = "loess", formula = y ~ x, se = F) +
  labs(title = "Residuals vs. Fitted",
       x = "Fitted Value of (1/a1c)", y = "Residual")

p2 <- ggplot(augi1, aes(sample = .std.resid)) +
  geom_qq() +
  geom_qq_line(col = "red") +
  labs(title = "Normal Q-Q plot",
       y = "Standardized Residual",
       x = "Standard Normal Quantiles")
```

Residual Plots for `imod_1` (via `ggplot2`)



Base R Residual Plots: `imod_1`

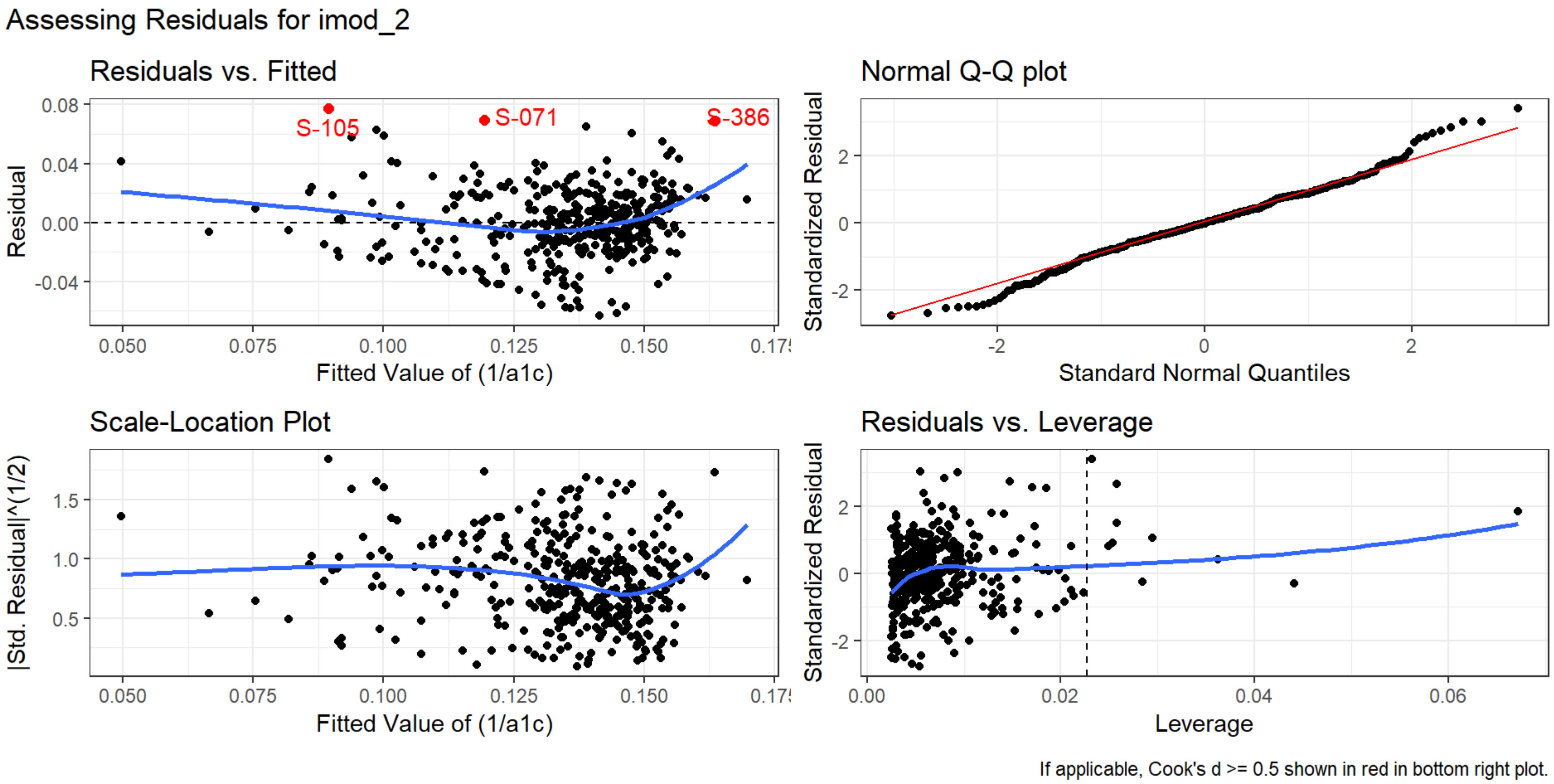
```r
par(mfrow = c(2,2)); plot(imod_1); par(mfrow = c(1,1))
```



Residual Plots for imod_2 (via ggplot2)

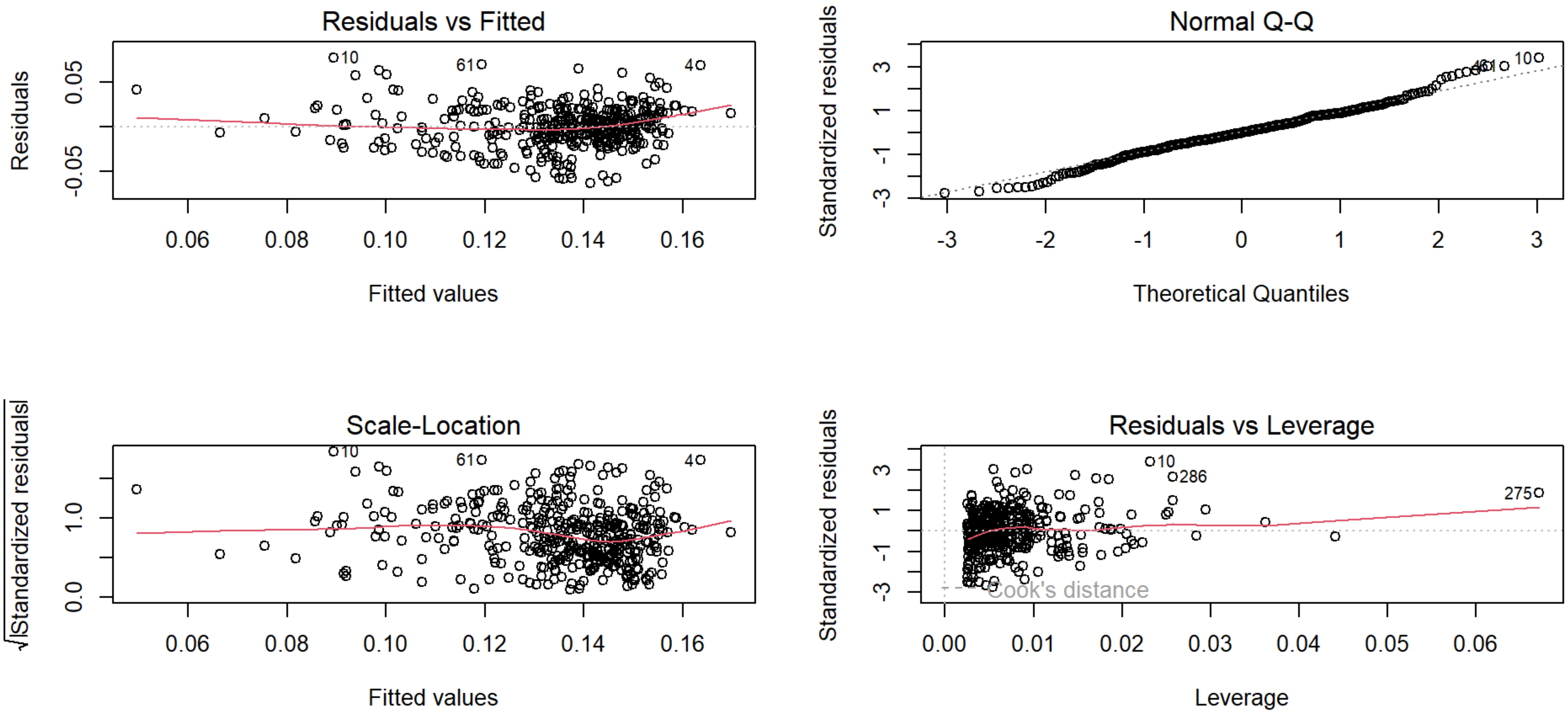
```r
p1 <- ggplot(augi2, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_point(data = augi2 |>
               slice_max(abs(.resid), n = 3),
             col = "red", size = 2) +
  geom_text_repel(data = augi2 |>
               slice_max(abs(.resid), n = 3),
               aes(label = subject), col = "red") +
  geom_abline(intercept = 0, slope = 0, lty = "dashed") +
  geom_smooth(method = "loess", formula = y ~ x, se = F) +
  labs(title = "Residuals vs. Fitted",
       x = "Fitted Value of (1/a1c)", y = "Residual")

p2 <- ggplot(augi2, aes(sample = .std.resid)) +
  geom_qq() +
  geom_qq_line(col = "red") +
  labs(title = "Normal Q-Q plot",
       y = "Standardized Residual",
       x = "Standard Normal Quantiles")
```

Residual Plots for `imod_2` (via `ggplot2`)



Base R Residual Plots: `imod_2`

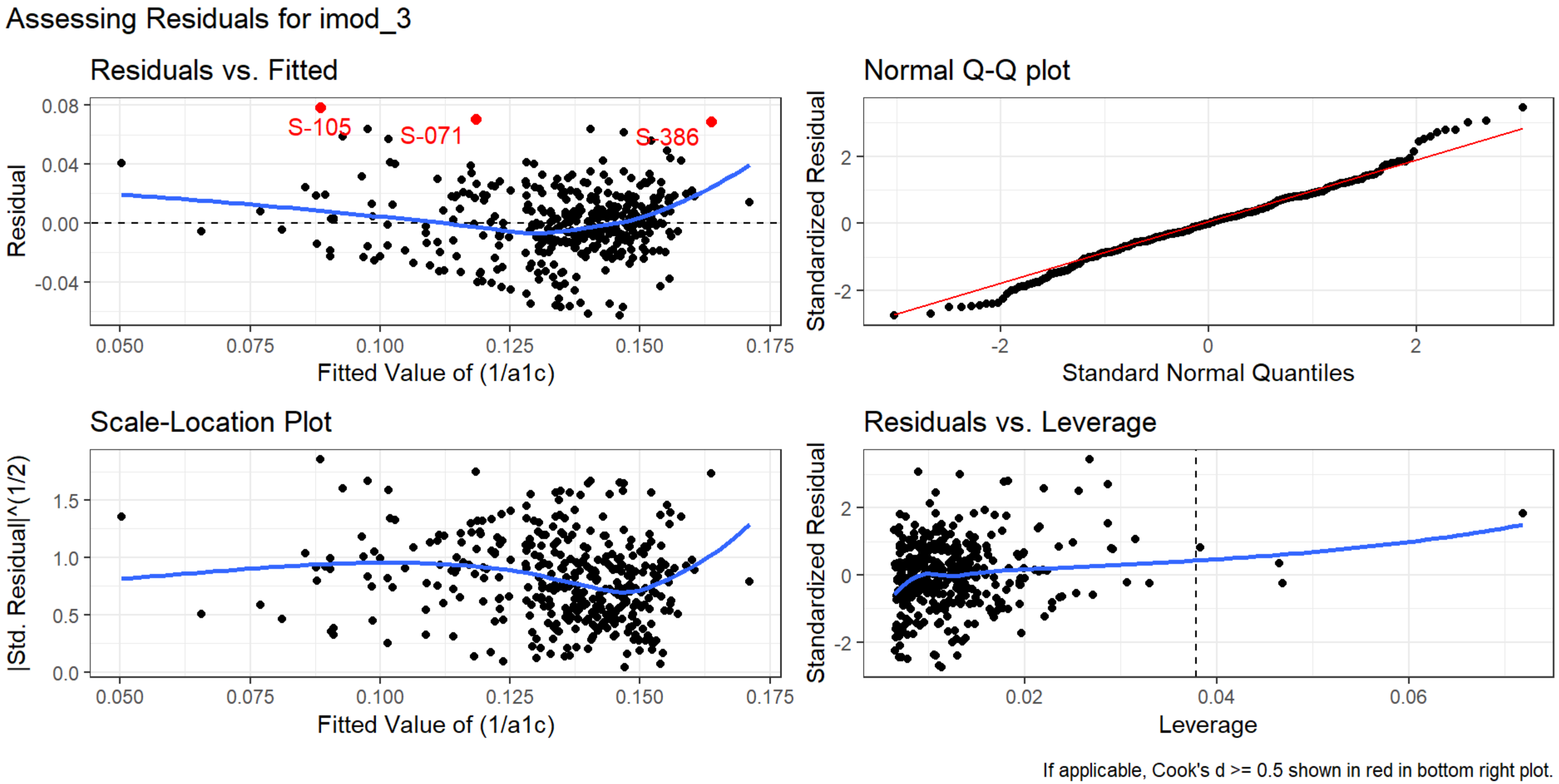
```r
par(mfrow = c(2,2)); plot(imod_2); par(mfrow = c(1,1))
```



Residual Plots for imod_3 (via ggplot2)

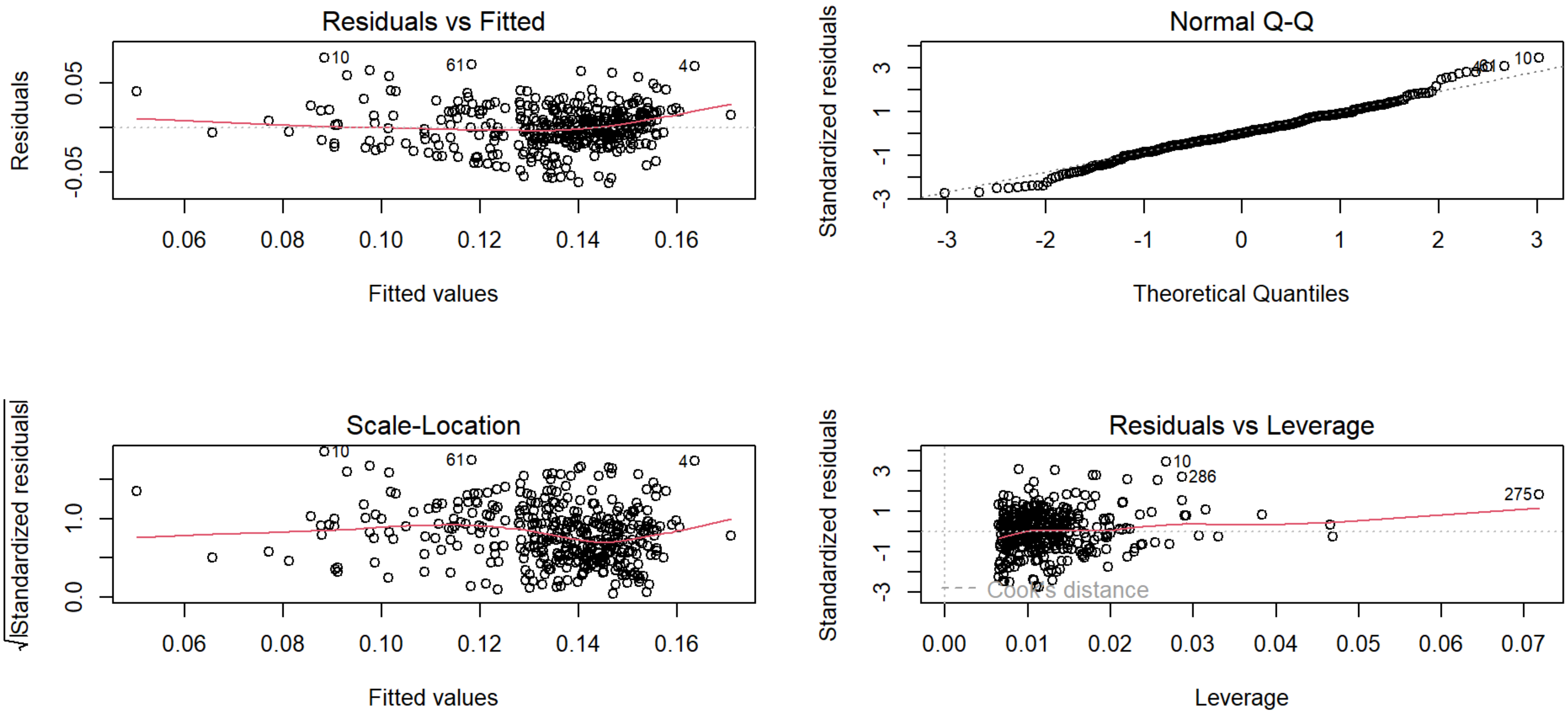
```r
p1 <- ggplot(augi3, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_point(data = augi3 |>
                  slice_max(abs(.resid), n = 3),
                col = "red", size = 2) +
  geom_text_repel(data = augi3 |>
                  slice_max(abs(.resid), n = 3),
                  aes(label = subject), col = "red") +
  geom_abline(intercept = 0, slope = 0, lty = "dashed") +
  geom_smooth(method = "loess", formula = y ~ x, se = F) +
  labs(title = "Residuals vs. Fitted",
       x = "Fitted Value of (1/a1c)", y = "Residual")

p2 <- ggplot(augi3, aes(sample = .std.resid)) +
  geom_qq() +
  geom_qq_line(col = "red") +
  labs(title = "Normal Q-Q plot",
       y = "Standardized Residual",
       x = "Standard Normal Quantiles")
```

Residual Plots for `imod_3` (via `ggplot2`)



Base R Residual Plots: imod_3

```r
par(mfrow = c(2,2)); plot(imod_3); par(mfrow = c(1,1))
```



Is collinearity a serious issue here?

```r
car::vif(imod_3)
```

```
             GVIF Df GVIF^(1/(2*Df))
a1c_old 1.025253  1        1.012548
age     1.047550  1        1.023499
income  1.041155  2        1.010134
```

None of these values exceed 5, so it doesn’t seem like there’s any problem.

```r
car::vif(imod_2)
```

```
 a1c_old      age 
1.017788 1.017788 
```

Conclusions so far (in-sample)?

1. In-sample model predictions are not wildly different in terms of accuracy across the three models.
   - Model `imod_3` has the best R^2, while
   - Model `imod_2` wins on adjusted R^2, σ and AIC, and
   - Model `imod_1` has the best BIC.
2. Residual plots look similarly reasonable for linearity, Normality and constant variance in all three models after imputation.

Calculate prediction errors in test samples

```r
test_im1 <- augment(imod_1, newdata = dm1_imp_test) |>
  mutate(name = "imod_1", fit_a1c = 1 / .fitted,
              res_a1c = a1c - fit_a1c)

test_im2 <- augment(imod_2, newdata = dm1_imp_test) |>
  mutate(name = "imod_2", fit_a1c = 1 / .fitted,
              res_a1c = a1c - fit_a1c)

test_im3 <- augment(imod_3, newdata = dm1_imp_test) |>
  mutate(name = "imod_3", fit_a1c = 1 / .fitted,
              res_a1c = a1c - fit_a1c)

test_icomp <- bind_rows(test_im1, test_im2, test_im3) |>
  arrange(subject, name)
```

Visualize Test-Sample Prediction Errors

```
p1 <- ggplot(test_icomp, aes(x = res_a1c, fill = name)) +
  geom_histogram(bins = 20, col = "white") +
  labs(x = "Prediction Errors on A1c scale", y = "") +
  facet_grid (name ~ .) + guides(fill = "none")

p2 <- ggplot(test_icomp, aes(x = factor(name), y = res_a1c,
                             fill = name)) +
  geom_violin(alpha = 0.3) +
  geom_boxplot(width = 0.3, notch = TRUE) +
  scale_x_discrete(position = "top",
                   limits =
                     rev(levels(factor(test_icomp$name)))) +
  guides(fill = "none") +
  labs(x = "", y = "Prediction Errors on A1c scale") +
  coord_flip()

p1 + p2 + plot_layout(ncol = 2)
```

Visualize Test-Sample Prediction Errors

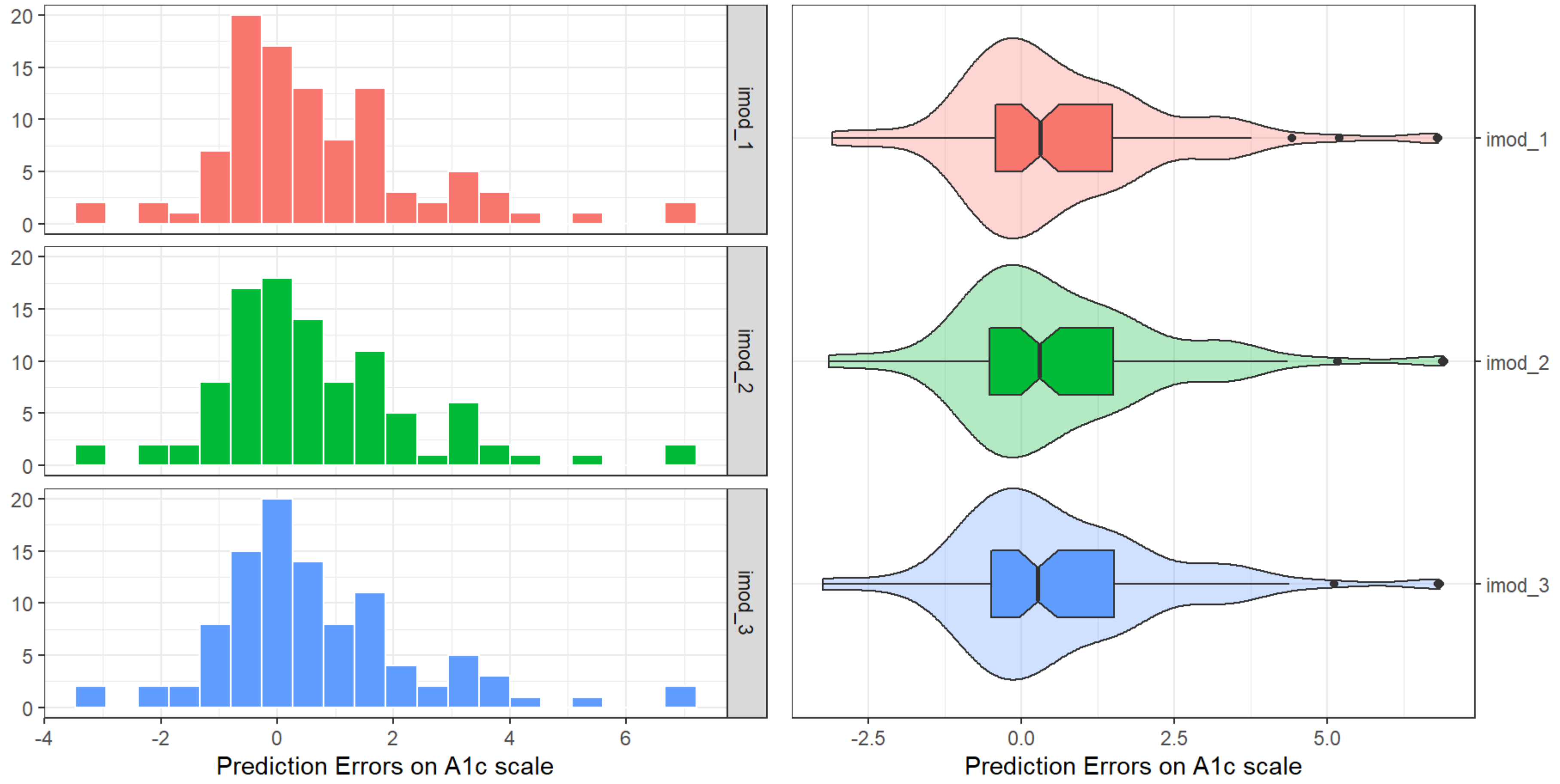


Table Comparing Model Prediction Errors

```
test_icomp |> group_by(name) |>
  summarize(n = n(), MAPE = mean(abs(res_a1c)), RMSPE = sqrt(mean(res_a1c^2
            max_error = max(abs(res_a1c))) |>
  kbl(digits = c(0, 0, 3, 3, 2)) |> kable_minimal(font_size = 28)
```

name	n	MAPE	RMSPE	max_error
imod_1	100	1.274	1.859	6.81
imod_2	100	1.282	1.864	6.91
imod_3	100	1.287	1.866	6.84

- Conclusions?

Identify the largest errors (Results)

Identify the subject(s) where that maximum prediction error was made by each model, and the observed and model-fitted values of a1c in each case.

```
tempi1 <- test_im1 |>
  filter(abs(res_a1c) == max(abs(res_a1c)))

tempi2 <- test_im2 |>
  filter(abs(res_a1c) == max(abs(res_a1c)))

tempi3 <- test_im3 |>
  filter(abs(res_a1c) == max(abs(res_a1c)))
```

```
# A tibble: 3 × 5
  subject name      a1c fit_a1c res_a1c
  <chr>   <chr>   <dbl>   <dbl>   <dbl>
1 S-282   imod_1     14    7.19    6.81
2 S-282   imod_2     14    7.09    6.91
3 S-282   imod_3     14    7.16    6.84
```

Line Plot of the Errors?

Compare the errors that are made at each level of observed A1c?

```
ggplot(test_icomp, aes(x = a1c, y = res_a1c,
                       group = name)) +
  geom_line(aes(col = name)) +
  geom_point(aes(col = name))
```

Key Summaries

With complete cases (from Classes 18-19)

- in-sample: all three models look OK on assumptions in residual plots, model 2 looks like it fits a little better by Adjusted R^2 and AIC, model 1 looks slightly better by BIC.
- out-of-sample: distributions of errors are similar. Model 1 has smallest MAPE, RMPSE and maximum error, while Model 2 has the smallest median error, but all three models are pretty similar.

Key Summaries

With imputation, (today)

- in-sample: nothing disastrous in residual plots, model 3 has the best R^2, Model 2 wins on adjusted R^2, σ, and AIC, and Model 1 has the best BIC.
- out-of-sample: Model 1 has the smallest MAPE, RMSE and maximum predictive error.

So what can we conclude? Does this particular imputation strategy have a big impact?

Again, this is our 431 Strategy

Which model is “most useful” in a prediction context?

1. Split the data into a model development (training) sample of about 70-80% of the observations, and a model test (holdout) 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.
4. Check adherence to regression assumptions in the development sample.
5. 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.)
6. Select a “final” model for use based on the evidence in steps 3, 4 and especially 5.

Clean Up

```r
rm(augi1, augi2, augi3,
   fwd.model, imod_0, imod_1, imod_2, imod_3,
   min.model, p1, p2, p3, p4, temp,
   tempi1, tempi2, tempi3,
   test_icomp, test_im1, test_im2, test_im3,
   tidy_im1, tidy_im2, tidy_im3)
```

Session Information

```r
sessionInfo()
```

```
R version 4.2.1 (2022-06-23 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 22000)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base
```