Exploring Frequency Models, AA ECF

Robert M Flight

## Purpose

Explore the definition of the model relating frequency to m/z and try to make sure we have it right, because getting it wrong is just not cool.

library(ScanCentricPeakCharacterization)  
library(ggplot2)  
library(ggforce)  
library(patchwork)  
library(flextable)  
knitr::opts\_chunk$set(dpi = 600)

## M/Z to Frequency Models

We will explore four different models relating frequency to m/z:

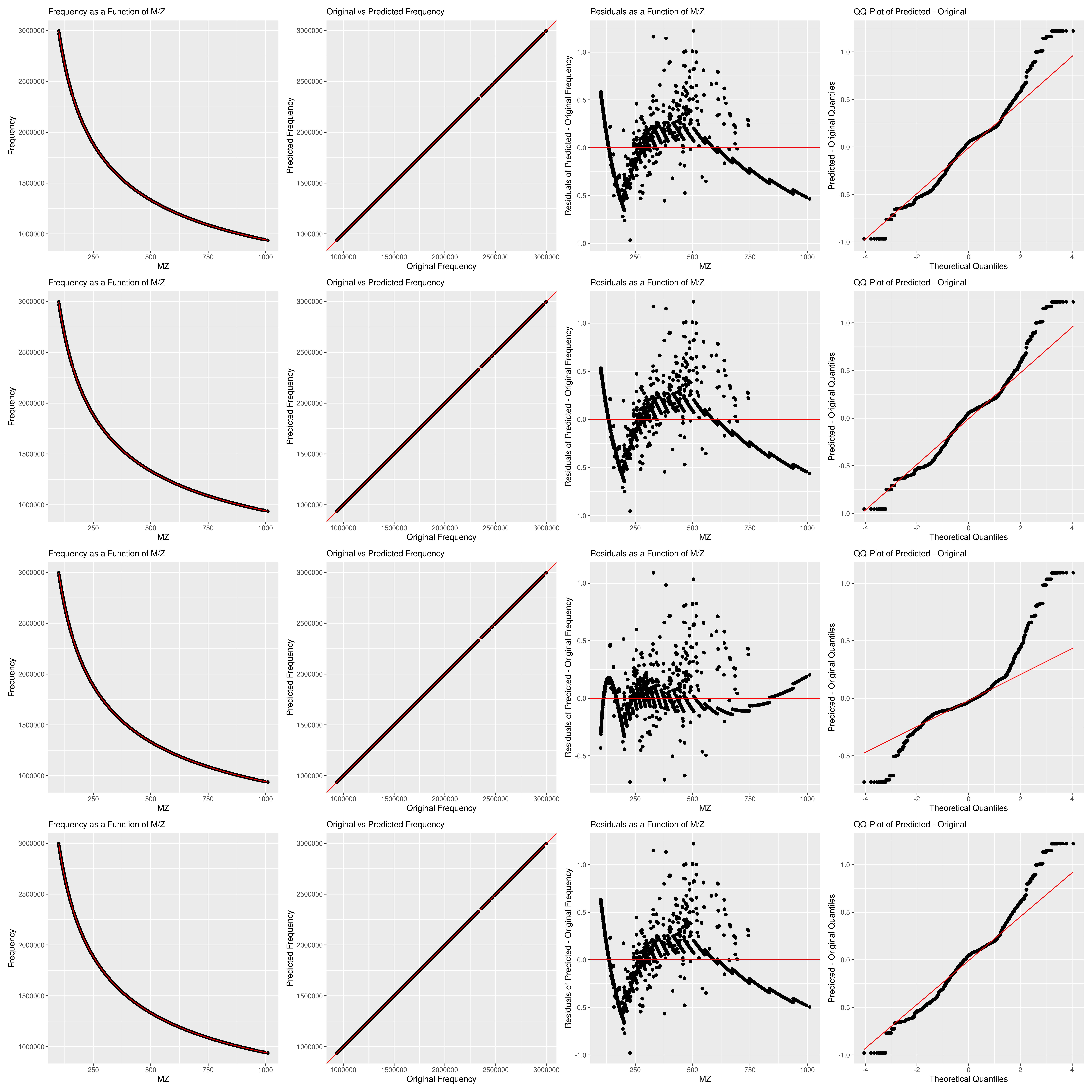
mod\_1 = c("a.freq" = 0, "y.freq" = -1/2)  
mod\_2 = c("a.freq" = 0, "x.freq" = -1, "y.freq" = -1/2)  
mod\_3 = c("a.freq" = 0, "x.freq" = -1, "y.freq" = -1/2, "z.freq" = -1/3)  
mod\_4 = c("a.freq" = 0, "y.freq" = -1/2, "z.freq" = -1/3)  
  
list\_models = list(mod1 = mod\_1,  
 mod2 = mod\_2,  
 mod3 = mod\_3,  
 mod4 = mod\_4)

We need to filter the scans first and figure out which ones to use. We are going to trust the third model for this.

sc\_raw = SCRaw$new("example\_data/161212\_unlabeledAAs\_1\_ECF.mzML")  
sc\_raw$extract\_raw\_data()  
  
sc\_raw$frequency\_fit\_description = mod\_3  
sc\_raw$predict\_frequency()  
mod\_3\_freq = sc\_raw$scan\_info  
use\_scans = (mod\_3\_freq$mad < 1)  
sc\_raw$raw\_df\_data = sc\_raw$raw\_df\_data[use\_scans]  
sc\_raw$scan\_info = sc\_raw$scan\_info[use\_scans, ]  
for (iname in names(mod\_3)) {  
 sc\_raw$scan\_info[[iname]] = NULL  
}

out\_models = purrr::map(list\_models, function(.x){  
 sc\_raw$frequency\_fit\_description = .x  
 sc\_raw$predict\_frequency()  
 tmp\_out = list(scans = sc\_raw$raw\_df\_data,  
 freq\_data = sc\_raw$scan\_info,  
 diag\_plots = sc\_raw$check\_frequency\_model(as\_list = TRUE))  
 for (iname in names(.x)) {  
 sc\_raw$scan\_info[[iname]] = NULL  
 }  
 tmp\_out  
})

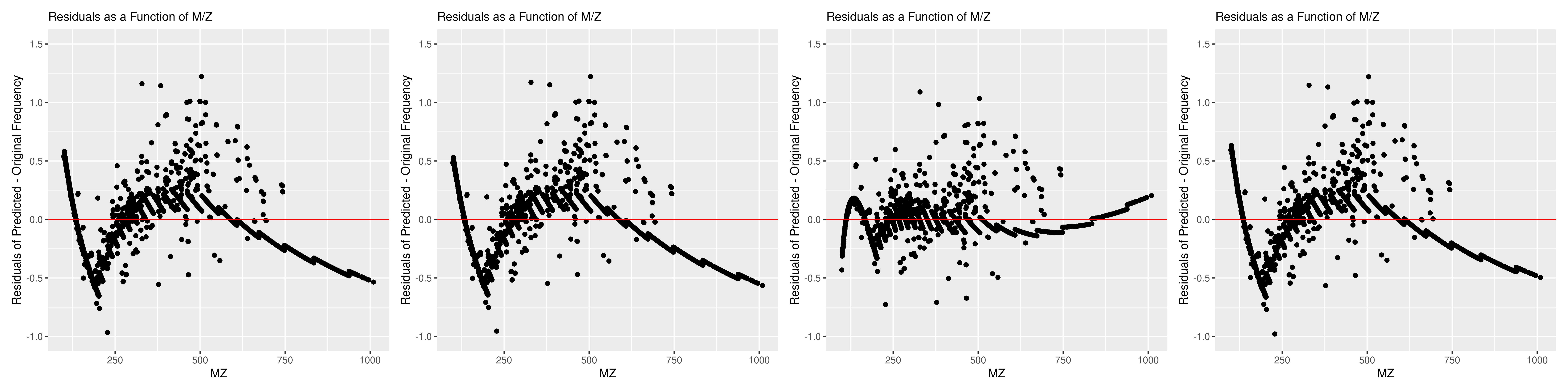
all\_plots = purrr::map(out\_models, function(in\_model){  
 tmp\_plot = wrap\_plots(in\_model$diag\_plots, nrow = 1)  
 tmp\_plot  
})  
  
all\_together = wrap\_plots(all\_plots, ncol = 1)  
  
all\_together



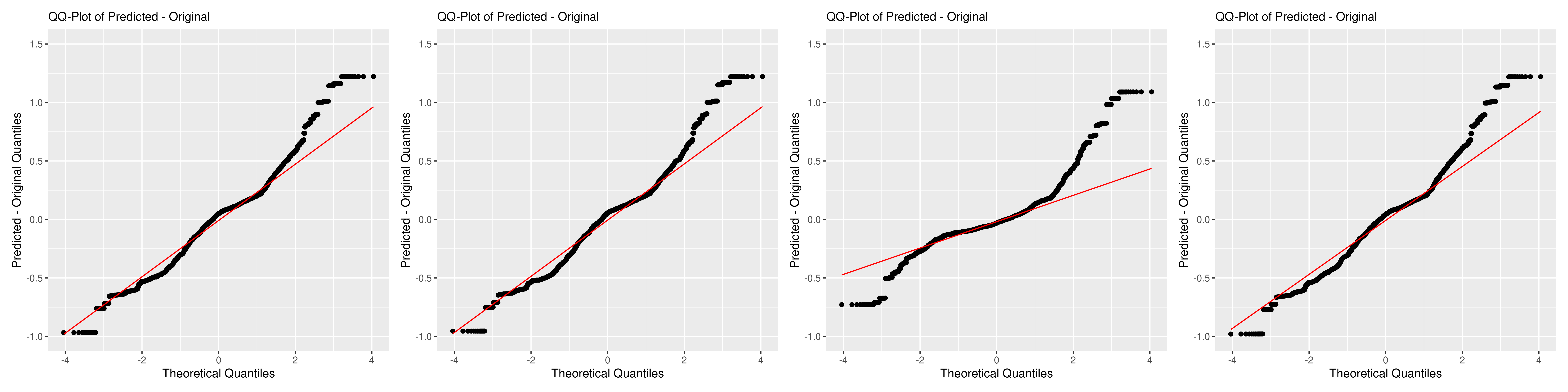
Looking at this, models 1 and 3 look like the best actual candidates. Of course, neither of those are the models we actually used for the Scan-Centric Peak Characterization manuscript.

We will try looking at these a little finer, especially at the distributions of the residuals and their QQ-Plots.

residual\_ylim = c(-1, 1.5)  
  
just\_residuals = purrr::map(out\_models, function(.x){  
 tmp\_plot = .x$diag\_plots$residuals\_as\_mz  
 tmp\_plot + coord\_cartesian(ylim = residual\_ylim)  
})  
  
wrap\_plots(just\_residuals, nrow = 1)



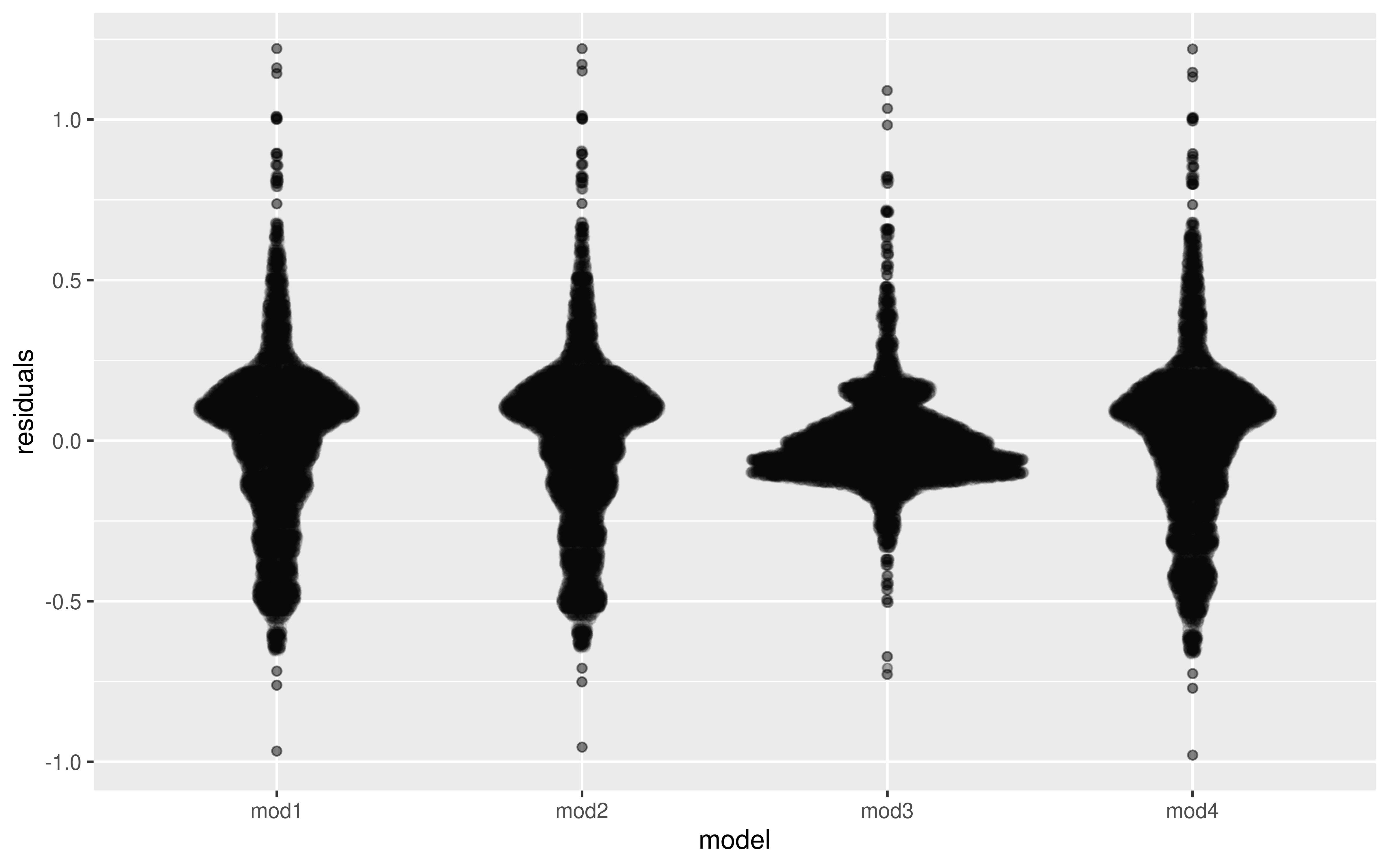
just\_qq = purrr::map(out\_models, function(.x){  
 tmp\_plot = .x$diag\_plots$qq\_residuals  
 tmp\_plot + coord\_cartesian(ylim = residual\_ylim)  
})  
wrap\_plots(just\_qq, nrow = 1)



Hmmm. after these two plots, I suspect it has something to do with how flat some of the residual patterns are instead of being curved.

One question we could ask is what are the distributions of those residuals like as a density or sina plot?

scan\_1\_residuals = purrr::imap\_dfr(out\_models, function(.x, .y){  
 tmp\_scan = .x$scans[[1]] %>%  
 dplyr::filter(convertable) %>%  
 dplyr::mutate(residuals = mean\_predicted,  
 model = .y)  
 tmp\_scan  
})  
  
scan\_1\_residuals %>%  
 ggplot(aes(x = model, y = residuals)) +  
 geom\_sina(alpha = 0.05)



Aha! A cautionary tale! The better QQ plot isn’t really **better** per se, because the bumps in the residuals actually make the distribution **fit** the normality assumption better. Lets look at the **MAD** for each of these.

scan\_1\_mads = purrr::imap\_dfr(out\_models, function(.x, .y){  
 #message(.y)  
 tmp\_info = .x$freq\_data %>%  
 dplyr::filter(scan %in% "s.157") %>%  
 dplyr::select(scan, median, mad) %>%  
 dplyr::mutate(model = .y)  
 tmp\_info  
})  
  
mad\_table = scan\_1\_mads %>%  
 flextable() %>%  
 colformat\_double(digits = 4)  
autofit(mad\_table)

| scan | median | mad | model |
| --- | --- | --- | --- |
| s.157 | 0.0512 | 0.2119 | mod1 |
| s.157 | 0.0556 | 0.2120 | mod2 |
| s.157 | -0.0306 | 0.1053 | mod3 |
| s.157 | 0.0430 | 0.2112 | mod4 |

OK, based on scan “s.157”, it’s all about **model 3**. Which again, is not the one we used, but it’s definitely one I understand we maybe should use.

Let’s double check these using plots of the medians and mad’s of the residuals in frequency across all scans for each of the models.

all\_mads = purrr::imap\_dfr(out\_models, function(.x, .y){  
 tmp\_info = .x$freq\_data %>%  
 dplyr::select(scan, median, mad) %>%  
 dplyr::mutate(model = .y)  
 tmp\_info  
})  
  
mad\_plot = all\_mads %>%  
 dplyr::select(mad, model) %>%  
 ggplot(aes(x = model, y = mad)) +  
 geom\_sina() +  
 labs(subtitle = "MADs")  
median\_plot = all\_mads %>%  
 dplyr::select(median, model) %>%  
 ggplot(aes(x = model, y = median)) +  
 geom\_sina() +  
 labs(subtitle = "Medians")  
mad\_plot | median\_plot



So this concludes again, we should be using **model 3** from above.

## Coefficients

Hunter was asking about the coefficients for each of these models wondering if one of them has smaller coefficients that lets an alternative one “bend” enough on it.

all\_coef = names(list\_models$mod3)  
coef\_df = as.data.frame(matrix(NA, nrow = 1, ncol = length(all\_coef)))  
names(coef\_df) = all\_coef  
  
get\_coef = function(curr\_coef, coef\_df){  
 match\_coef = names(coef\_df)[names(coef\_df) %in% names(curr\_coef)]  
 tmp\_coef = coef\_df  
 for (icoef in match\_coef) {  
 tmp\_coef[[icoef]] = curr\_coef[[icoef]][1]  
 }  
 tmp\_coef  
}  
  
model\_coefficients = purrr::imap\_dfr(out\_models, function(in\_model, id){  
 out\_coef = get\_coef(in\_model$freq\_data, coef\_df)  
 out\_coef$model = id  
 out\_coef  
})  
  
model\_coefficients %>%  
 flextable() %>%  
 colformat\_double(digits = 0) %>%  
 autofit()

| a.freq | x.freq | y.freq | z.freq | model |
| --- | --- | --- | --- | --- |
| 4 |  | 29,803,537 |  | mod1 |
| 4 | 45 | 29,803,531 |  | mod2 |
| -39 | 6,218 | 29,800,524 | 1,314 | mod3 |
| 3 |  | 29,803,518 | 11 | mod4 |

Based on this table, I don’t think that is what is going on.