

Linear Mixed Models - Questions

Lance Stasinski

9/27/2021

Linear Mixed Models - Questions

This document walks through the full process of creating linear mixed models with the lichen spectroscopy data, and the questions I currently have can be found in bold along the way.

Research Questions

1. How do reflectance measurements taken from a lichen thallus change as lichen specimens age?
 - Do lichen show a general pattern? i.e. Do the underlying traits change in a consistent manner between lichen species, such that only the starting values differentiate species rather than the rates of change for those corresponding traits.
 - Are there differences in the rate of reflectance change between species? I'm not concerned with exactly how each species responds to age.
2. Where does much of the variation in either the rate of reflectance change or the starting reflectance values arise? i.e. Does much of the variation occur between species or at higher taxonomic ranks such as class?

Linear mixed models appear to be a good choice for answering these questions.

Data structure

Full-range spectra (400 to 2400nm) are available at 1nm resolution for 30 species that represent 19 families, 16 orders, and 6 classes. The spectra per each species were taken from a type II chronosequence (trading space for time) of aging lichen thalli. Each thallus is represented by 4 reflectance measurements from various parts of the thallus to capture spectral variation (lichen surfaces can be quite heterogeneous!). The amount of data is not equal among species or higher taxonomic ranks or equal across the time scale (imbalanced).

Number of measurements per species

```
setwd("~/GitHub/Lichen-Herbarium-Spectra")  
  
library(spectrolab)
```

```
## Warning: package 'spectrolab' was built under R version 4.1.1
```

```
## spectrolab version: 0.0.14
##
## Please cite:
## Meireles J, Schweiger A, Cavender-Bares J (2017). spectrolab: Class
## and Methods for Spectral Data in R. doi: 10.5281/zenodo.3934575 (URL:
## https://doi.org/10.5281/zenodo.3934575), R package version 0.0.14,
## <URL: https://CRAN.R-project.org/package=spectrolab>.DOI: https://doi.org/10.5281/zenodo.3934575
```

```
##
## Attaching package: 'spectrolab'
```

```
## The following objects are masked from 'package:stats':
##
## sd, smooth, var
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:spectrolab':
##
## combine
```

```
## The following objects are masked from 'package:stats':
##
## filter, lag
```

```
## The following objects are masked from 'package:base':
##
## intersect, setdiff, setequal, union
```

```
spectra = readRDS('spectra/lichen_spectra.rds')
spec_df = as.data.frame(spectra)
spec_df %>% count(scientificName)
```

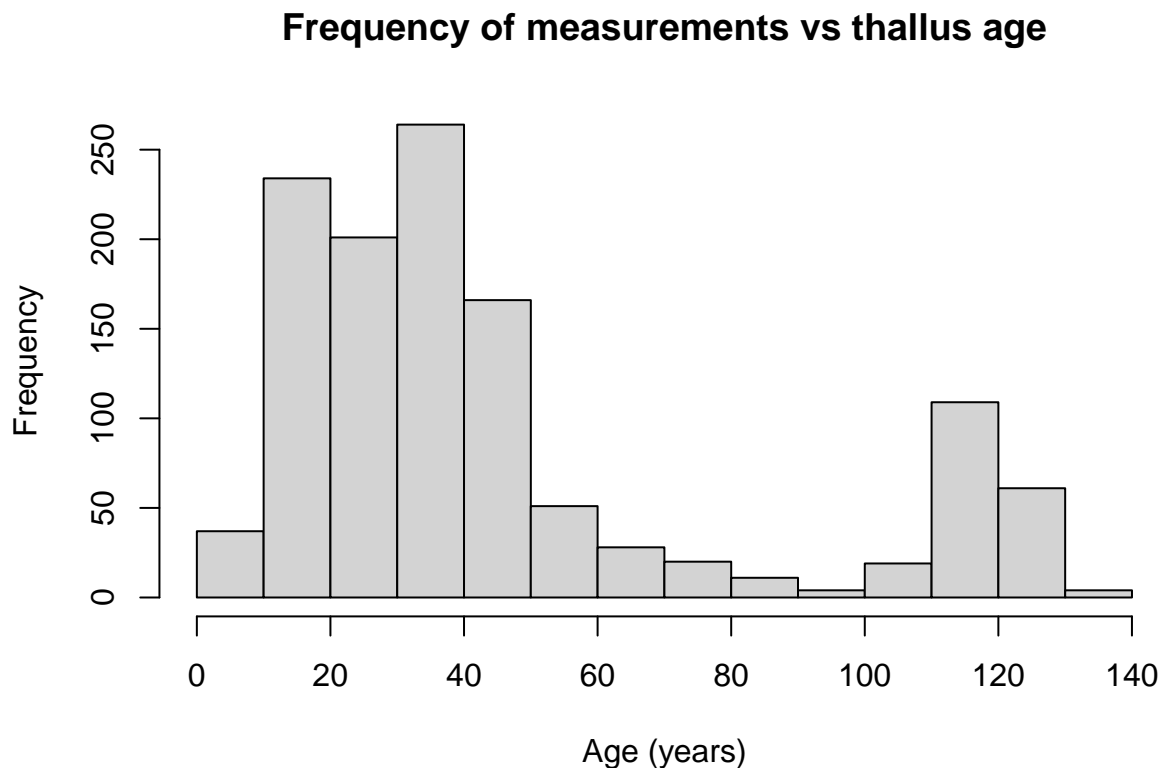
```
##           scientificName    n
## 1   Acarospora_americana  32
## 2   Baeomyces_rufus      61
## 3   Caloplaca_flavovirescens 60
## 4   Candelaria_concolor  62
## 5   Chrysothrix_candelaris 52
## 6   Dimelaena_oreina     60
## 7   Ephebe_ocellata     24
## 8   Flavoparmelia_baltimorensis 17
## 9   Flavoparmelia_caperata 119
## 10  Flavoparmelia_euplecta   8
## 11  Flavoparmelia_haysomii  20
## 12  Flavoparmelia_rutidota   4
## 13  Flavoparmelia_soredians   8
## 14  Flavopunctelia_darrowii   4
```

| | | |
|-------|---------------------------|----|
| ## 15 | Flavopunctelia_flaventior | 12 |
| ## 16 | Flavopunctelia_praesignis | 21 |
| ## 17 | Flavopunctelia_soredica | 16 |
| ## 18 | Graphis_scripta | 44 |
| ## 19 | Ionaspis_lacustris | 52 |
| ## 20 | Lecidea_tessellata | 44 |
| ## 21 | Loxospora_elatina | 56 |
| ## 22 | Neofuscelia_verruculifera | 4 |
| ## 23 | Peltigera_elisabethae | 68 |
| ## 24 | Pertusaria_opthalmiza | 64 |
| ## 25 | Rhizocarpon_grande | 55 |
| ## 26 | Strigula_submuriformis | 48 |
| ## 27 | Trypethelium_virens | 79 |
| ## 28 | Umbilicaria_muehlenbergii | 48 |
| ## 29 | Verrucaria_fuscella | 51 |
| ## 30 | Xanthoparmelia_darrowii | 16 |

Note: This table shows the number of measurements per species, not number of individual thalli.

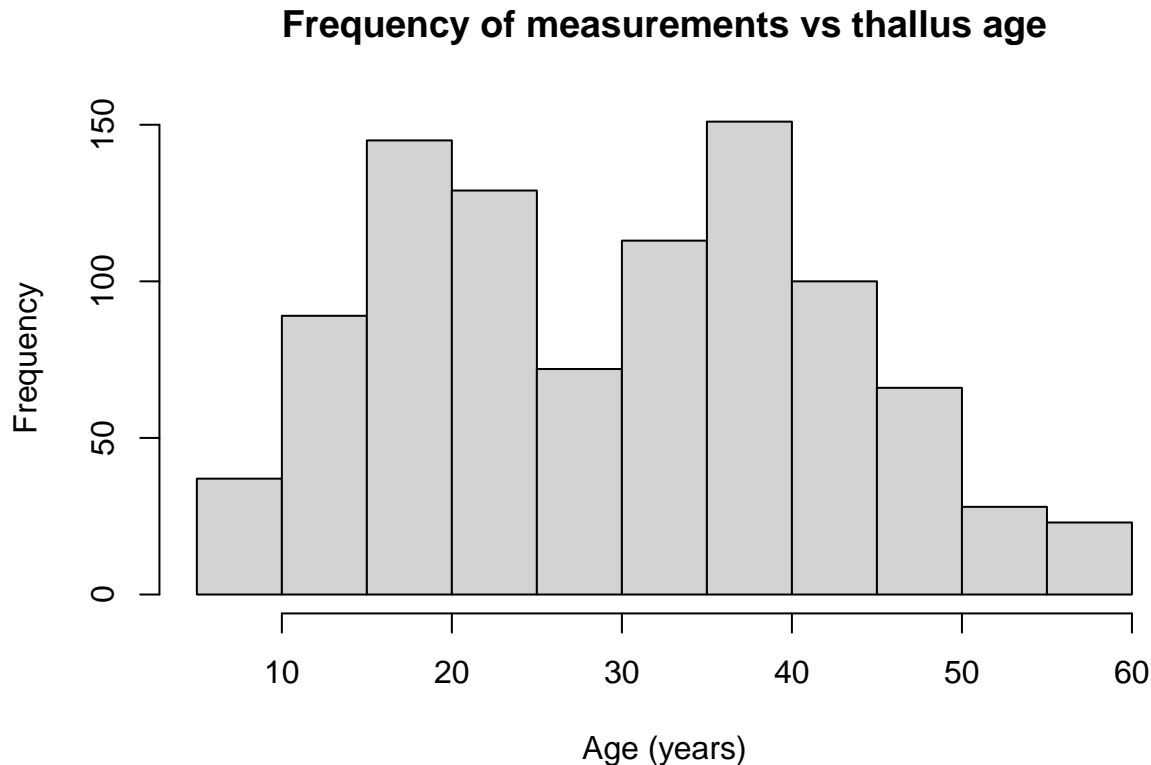
Histogram of measurements across speciemn age

```
hist(spec_df$age, main = 'Frequency of measurements vs thallus age', xlab = 'Age (years)')
```



Clearly there is an issue with bimodal distribution of age, so I will elect to limit the dataset to only measurements taken on specimens 60 years old and younger.

```
spec_df = spec_df[spec_df$age <= 60, ]
hist(spec_df$age, main = 'Frequency of measurements vs thallus age', xlab = 'Age (years)')
```



This seems a little bit better for understanding how reflectance changes as thalli age.

Question 1: Does this age range seem appropriate? Should the age range be trimmed further? Perhaps a range of 10-50 years old would allow for more balance in the data (there would be loss of information on the early part of the aging process, yet there's not much to begin with).

Model Assumptions

Do the data fit the assumptions required by linear mixed models/the underlying linear model? I will use 3 wavelengths, 1 from each of the three spectral regions (VIS, NIR, SWIR), to get a rough estimate as to how the data fit the model assumptions. I'll use wavelengths 550, 850, and 1550 because they are near the middle of their respective spectral regions and are not part of major or minor waterbands which could potentially throw off estimates.

The underlying OLS model assumptions

1. Independence - The individual measurements from single thallus are NOT independent. Thus, it may make more sense to reduce the data to the mean reflectance per thallus. Further, the reflectance between individual thalli are not independent within a species - they share evolutionary history. However, this should be fixed by treating species as a random effect in the linear mixed model.

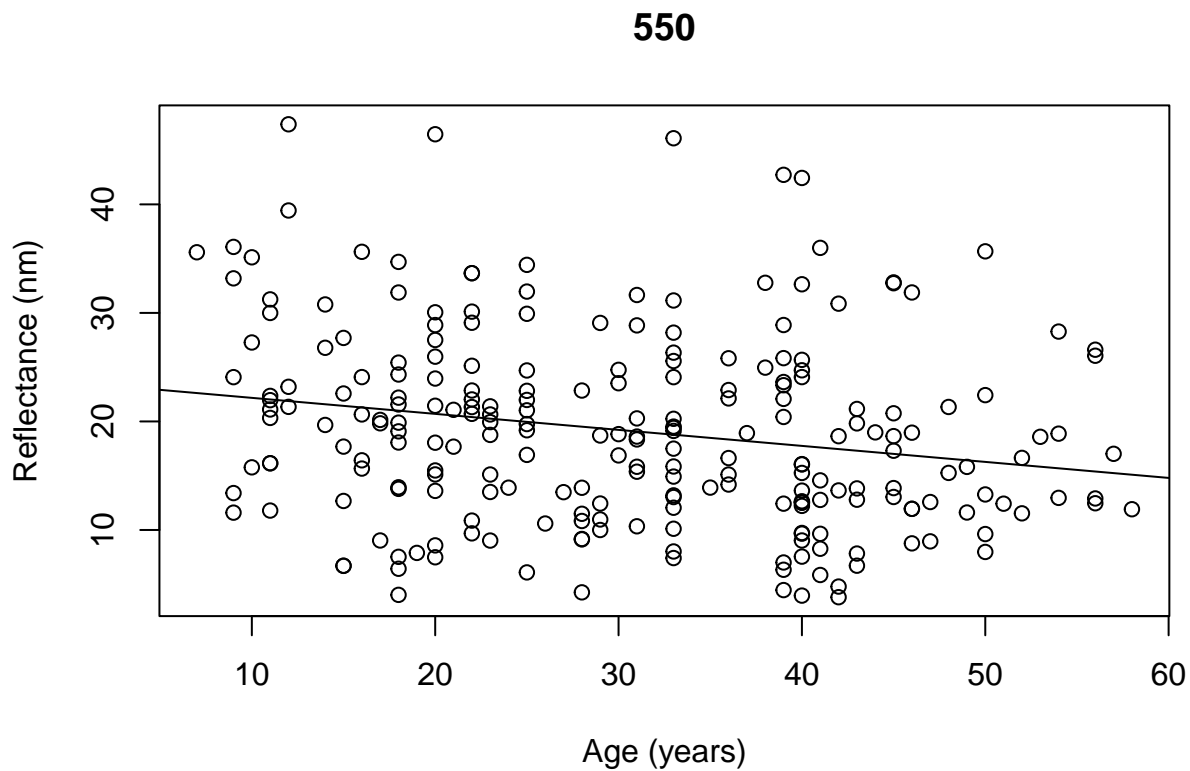
Question 2: Is using the mean spectra acceptable? Should I account for the variation in the spectra per individual in some way?

```
spectra = spectra[meta(spectra)$age < 60,]
spectra = aggregate(spectra, meta(spectra)$X, mean, try_keep_txt(mean)) #X indicates individual thallus
data = meta(spectra)
spec.m = as.matrix(spectra) * 100 #convert reflectance to a percentage to help with interpretation
spectra_percent = as_spectra(spec.m)
meta(spectra_percent) = data
spec_df = as.data.frame(spectra_percent)
```

2. Linearity

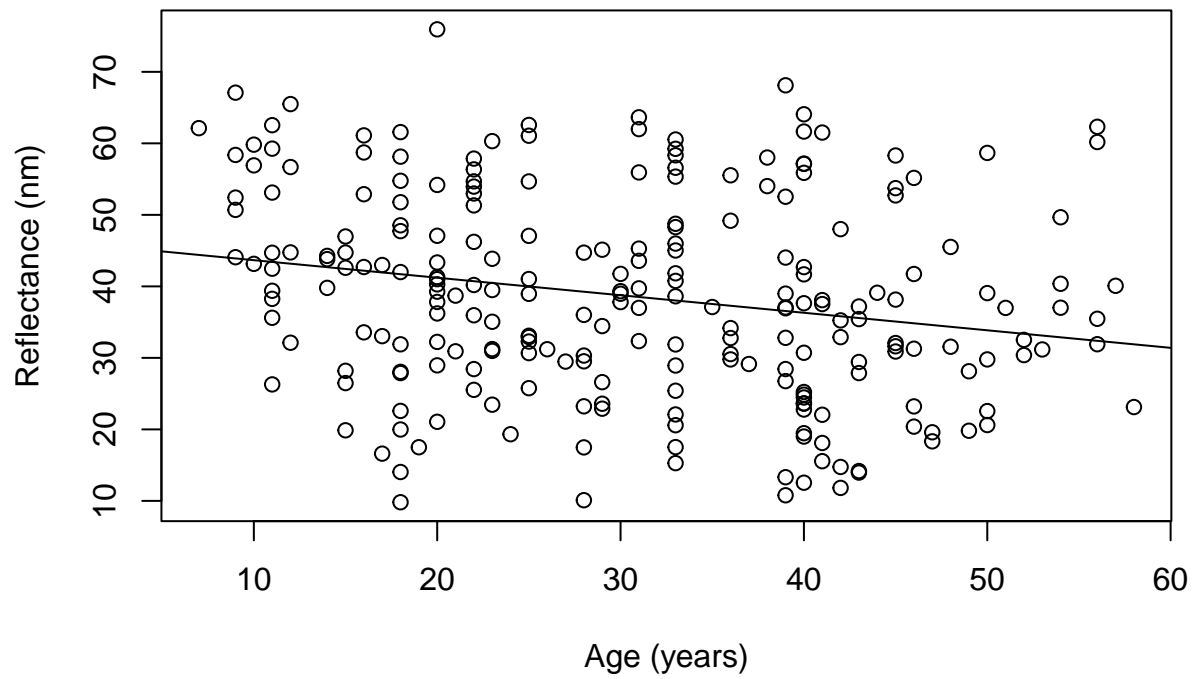
```
lm550 = lm(spec_df[, '550'] ~ spec_df$age)
lm850 = lm(spec_df[, '850'] ~ spec_df$age)
lm1550 = lm(spec_df[, '1550'] ~ spec_df$age)

plot(spec_df$age, spec_df[, '550'], main = '550', ylab = 'Reflectance (nm)', xlab = 'Age (years)')
abline(lm550)
```



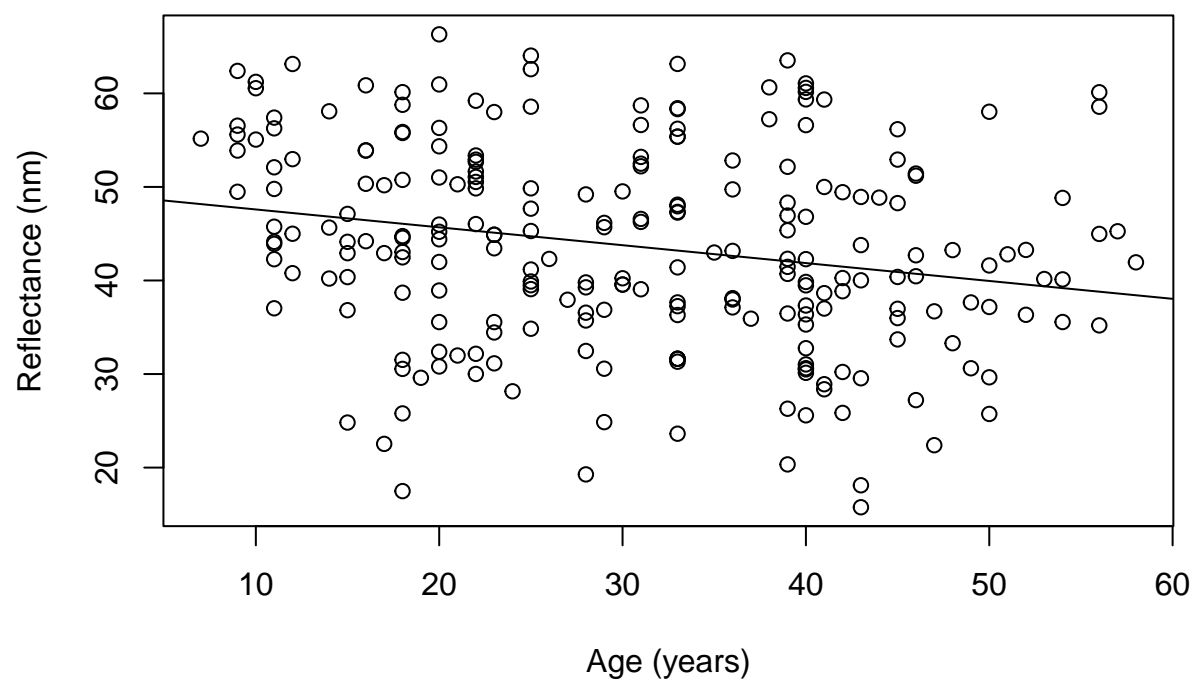
```
plot(spec_df$age, spec_df[, '850'], main = '850', ylab = 'Reflectance (nm)', xlab = 'Age (years)')
abline(lm850)
```

850



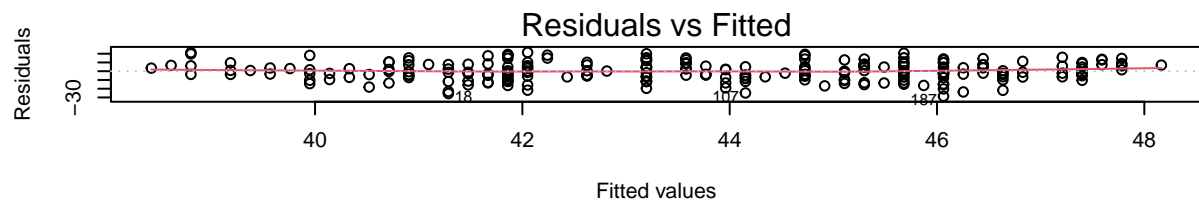
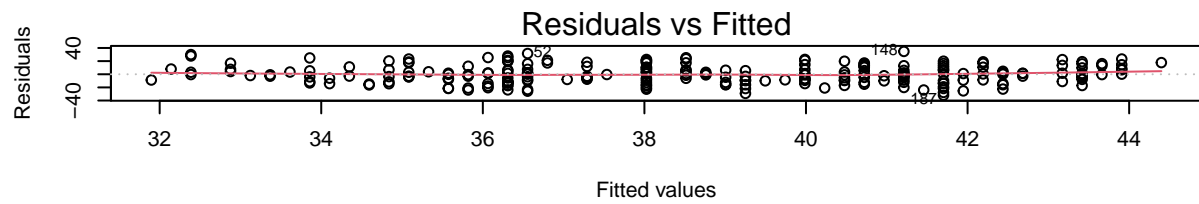
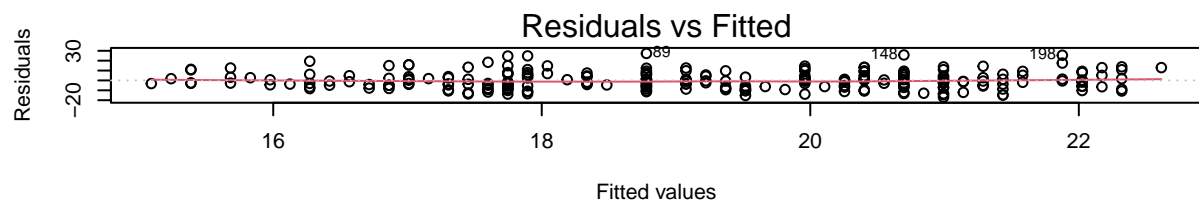
```
plot(spec_df$age, spec_df[, '1550'], main = '1550', ylab = 'Reflectance (nm)', xlab = 'Age (years)')  
abline(lm1550)
```

1550



It seems like linearity is met. At least it does not look like any other function type would fit the data better.

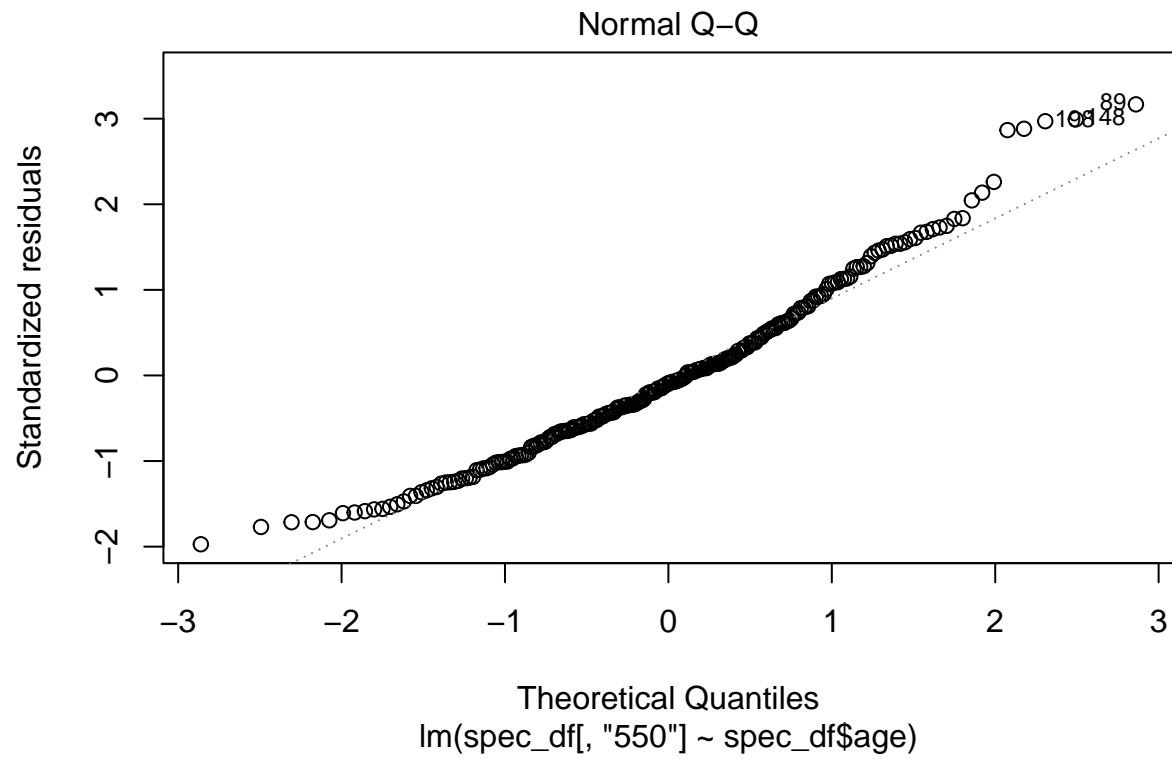
```
par(mfrow = c(3,1))  
plot(lm550, 1)  
plot(lm850, 1)  
plot(lm1550, 1)
```



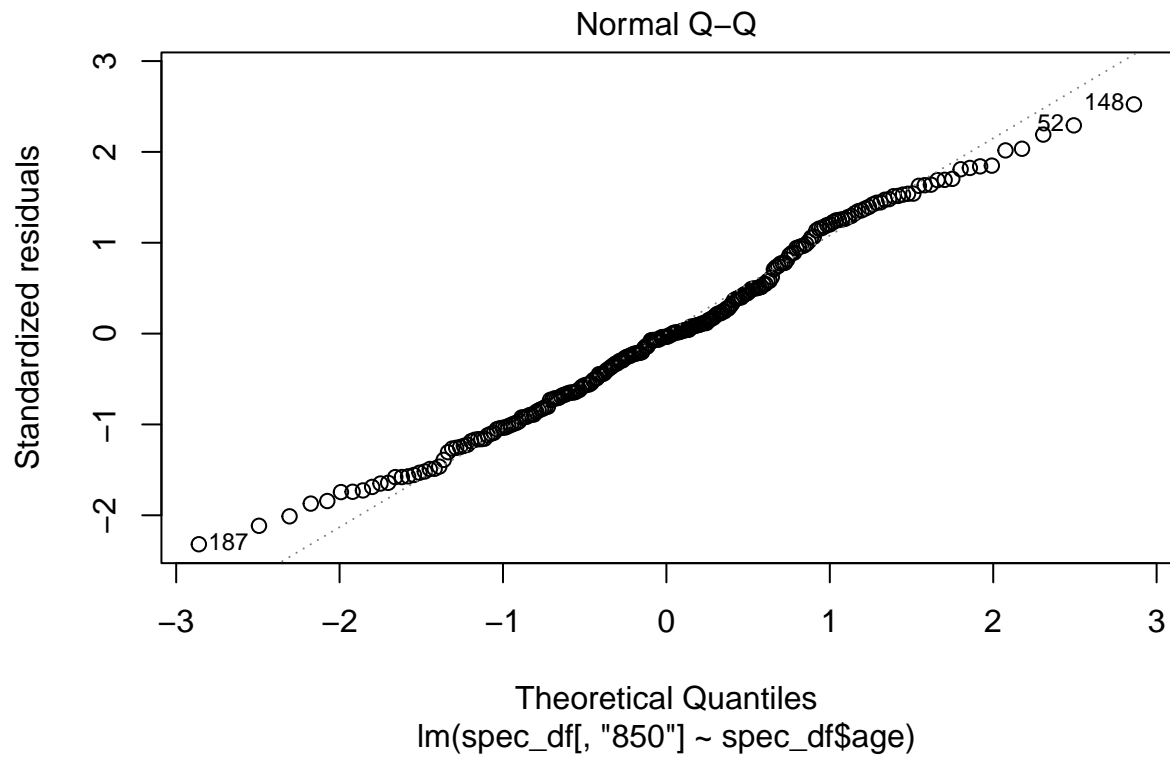
The residuals versus fitted plots also indicate that a linear function is a decent fit for the data.

3. Normality

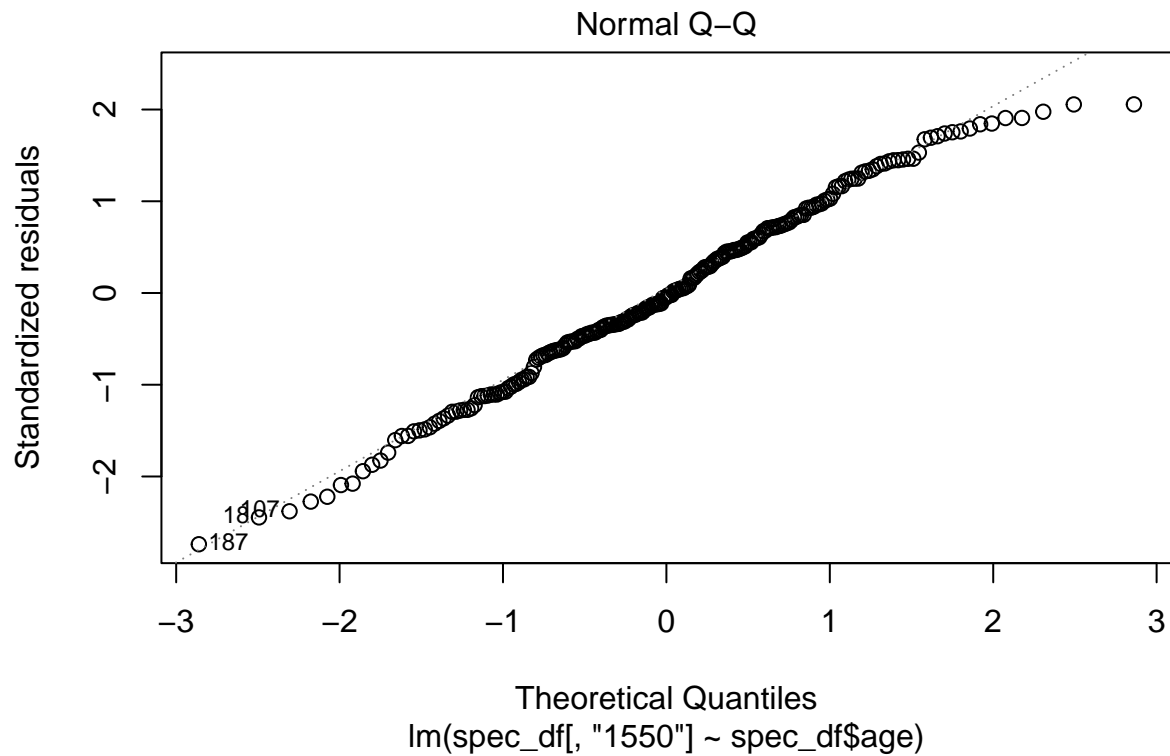
```
plot(lm550, 2)
```

```
plot(lm850, 2)
```



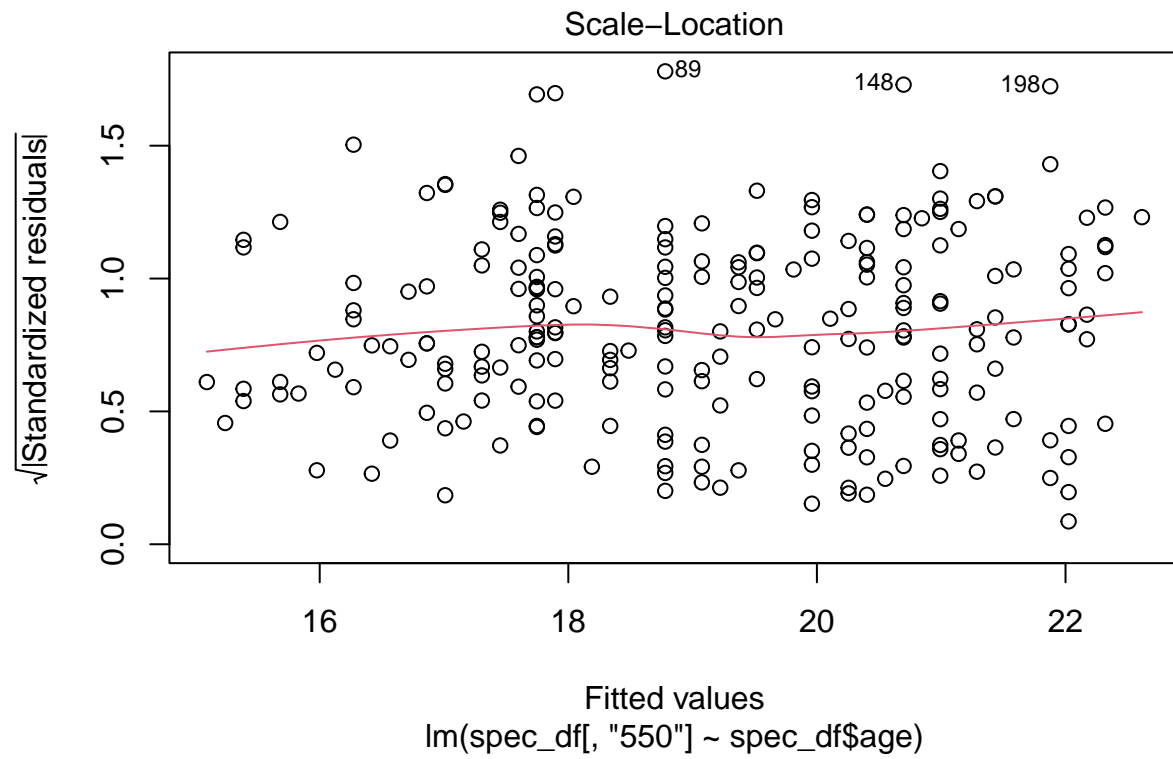
```
plot(lm1550, 2)
```



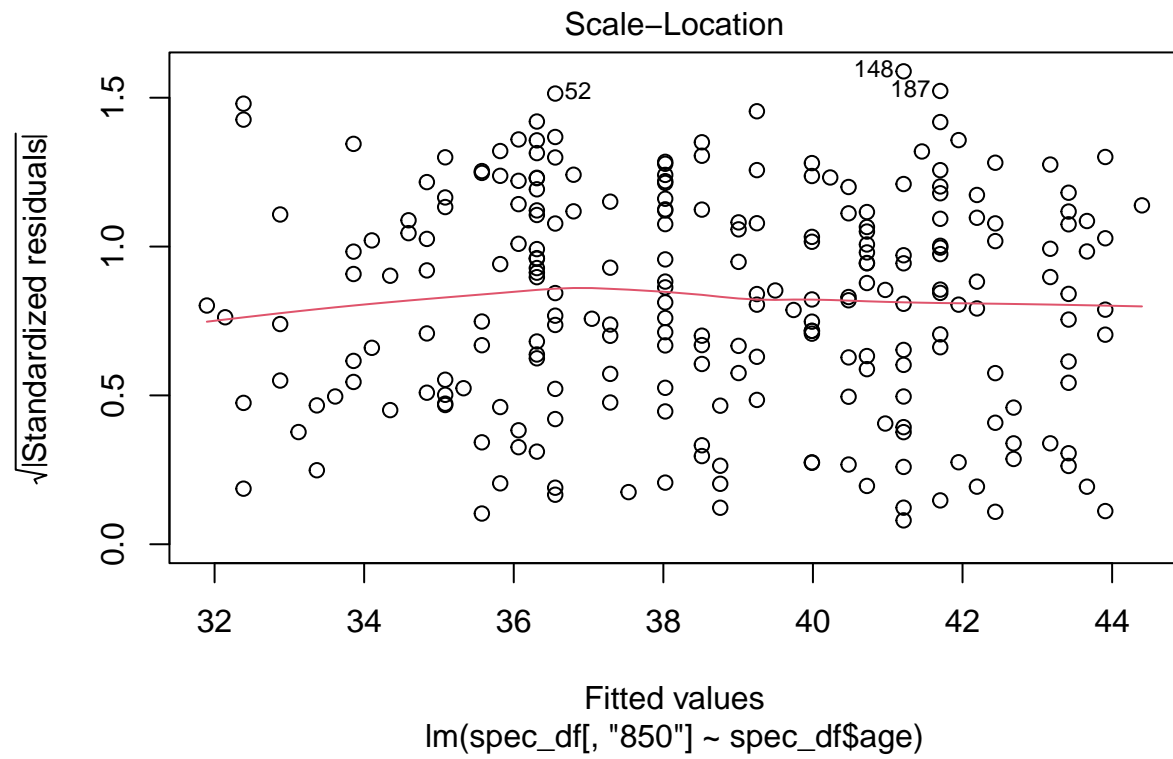
The tails are a bit skewed from the reference line, but it seems like the residuals are close to normally distributed.

4. Homoscedasticity

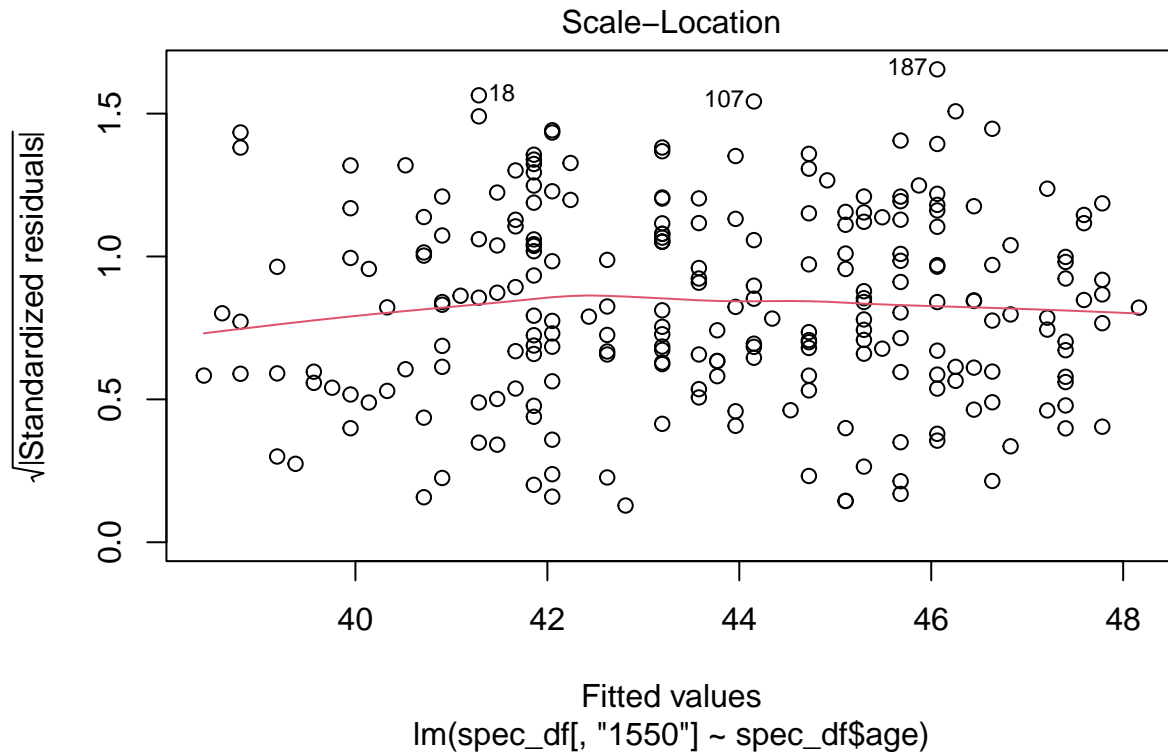
```
plot(lm550, 3)
```



```
plot(lm850, 3)
```



```
plot(lm1550, 3)
```



Looks pretty homoscedastic to me.

Overall, it looks like the assumptions of the linear model are generally met.

Linear Mixed Model Assumptions

Before checking model assumptions, let's create a linear mixed model by treating species (denoted as `scientificName` in the dataframe) as a random effect. Let's also compare a variable intercept - fixed slope model to a variable intercept - variable slope model for each of the selected 3 wavelengths.

```
library(lme4)

## Loading required package: Matrix

varInt550 = lmer(spec_df[, '550'] ~ age + (1|scientificName),
                 data = spec_df, REML = T,
                 lmerControl(optimizer='bobyqa', #prevents convergence error
                             boundary.tol = 1e-5, optCtrl = list(maxfun = 1e5)))
varSlope550 = lmer(spec_df[, '550'] ~ age + (1+age|scientificName),
                  data = spec_df, REML = T,
                  lmerControl(optimizer='bobyqa',
                              boundary.tol = 1e-5, optCtrl = list(maxfun = 1e5)))
anova(varInt550, varSlope550)

## refitting model(s) with ML (instead of REML)
```

```
## Data: spec_df
## Models:
## varInt550: spec_df[, "550"] ~ age + (1 | scientificName)
## varSlope550: spec_df[, "550"] ~ age + (1 + age | scientificName)
##           npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## varInt550      4 1510.5 1524.4 -751.27   1502.5
## varSlope550    6 1514.3 1535.1 -751.14   1502.3 0.262  2    0.8772
```

```
library(lme4)
varInt850 = lmer(spec_df[, '850'] ~ age + (1|scientificName),
                 data = spec_df, REML = T,
                 lmerControl(optimizer = 'bobyqa', #prevents convergence error
                             boundary.tol = 1e-5, optCtrl = list(maxfun = 1e5)))
varSlope850 = lmer(spec_df[, '850'] ~ age + (1+age|scientificName),
                  data = spec_df, REML = T,
                  lmerControl(optimizer = 'bobyqa',
                              boundary.tol = 1e-5, optCtrl = list(maxfun = 1e5)))
```

```
## boundary (singular) fit: see ?isSingular
```

```
anova(varInt850, varSlope850)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: spec_df
## Models:
## varInt850: spec_df[, "850"] ~ age + (1 | scientificName)
## varSlope850: spec_df[, "850"] ~ age + (1 + age | scientificName)
##           npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## varInt850      4 1679.2 1693.1 -835.6   1671.2
## varSlope850    6 1683.2 1704.0 -835.6   1671.2 0.0018  2    0.9991
```

```
library(lme4)
varInt1550 = lmer(spec_df[, '1550'] ~ age + (1|scientificName),
                  data = spec_df, REML = T,
                  lmerControl(optimizer = 'bobyqa', #prevents convergence error
                              boundary.tol = 1e-5, optCtrl = list(maxfun = 1e5)))
varSlope1550 = lmer(spec_df[, '1550'] ~ age + (1+age|scientificName),
                    data = spec_df, REML = T,
                    lmerControl(optimizer = 'bobyqa',
                                boundary.tol = 1e-5, optCtrl = list(maxfun = 1e5)))
```

```
## boundary (singular) fit: see ?isSingular
```

```
anova(varInt1550, varSlope1550)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: spec_df
## Models:
```

```
## varInt1550: spec_df[, "1550"] ~ age + (1 | scientificName)
## varSlope1550: spec_df[, "1550"] ~ age + (1 + age | scientificName)
##           npar      AIC   BIC logLik deviance  Chisq Df Pr(>Chisq)
## varInt1550      4 1621.2 1635 -806.58   1613.2
## varSlope1550     6 1625.2 1646 -806.58   1613.2 0.0035  2      0.9983
```

NOTE: The boundary (singular) fit: see `?isSingular` warning arises from the variable intercept - variable slope models. - Indicates “some dimensions of the variance-covariance matrix have been estimated as exactly zero.”

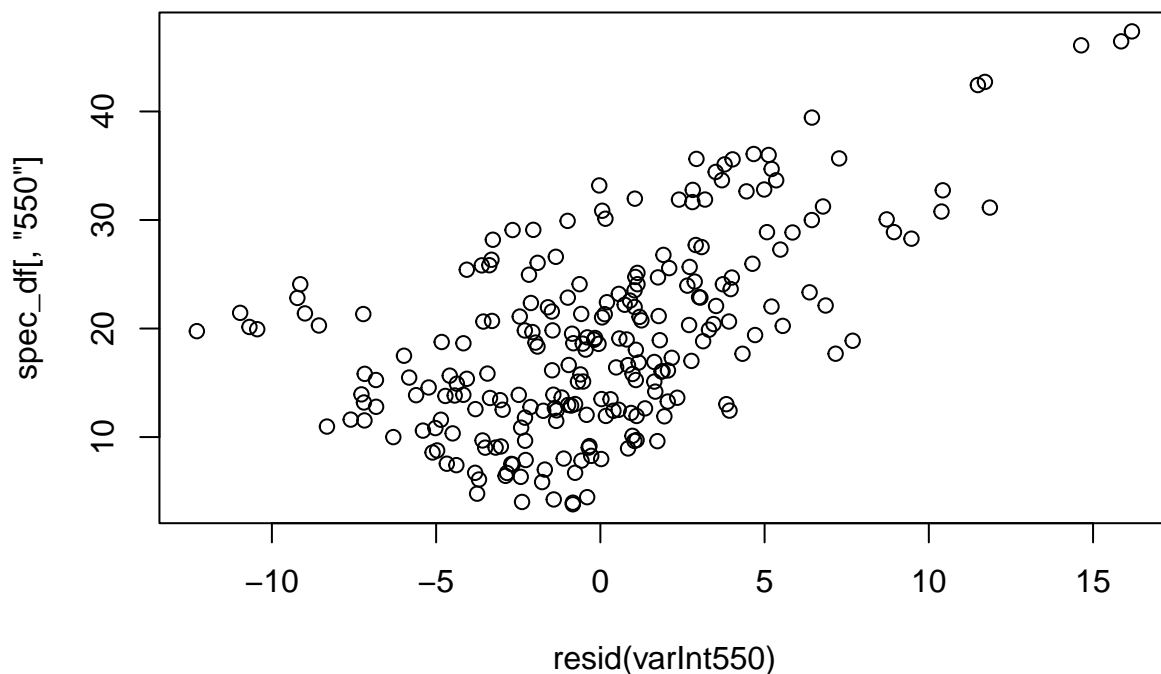
Question 3: Is a singular fit something I need to fix? If so, how? I’m not finding any clear answers on any blogs/documentation.

It looks like the variable intercept - fixed slope model is a better fit for each of these three wavelengths. This also turns out to be true for most wavelengths (not presented here).

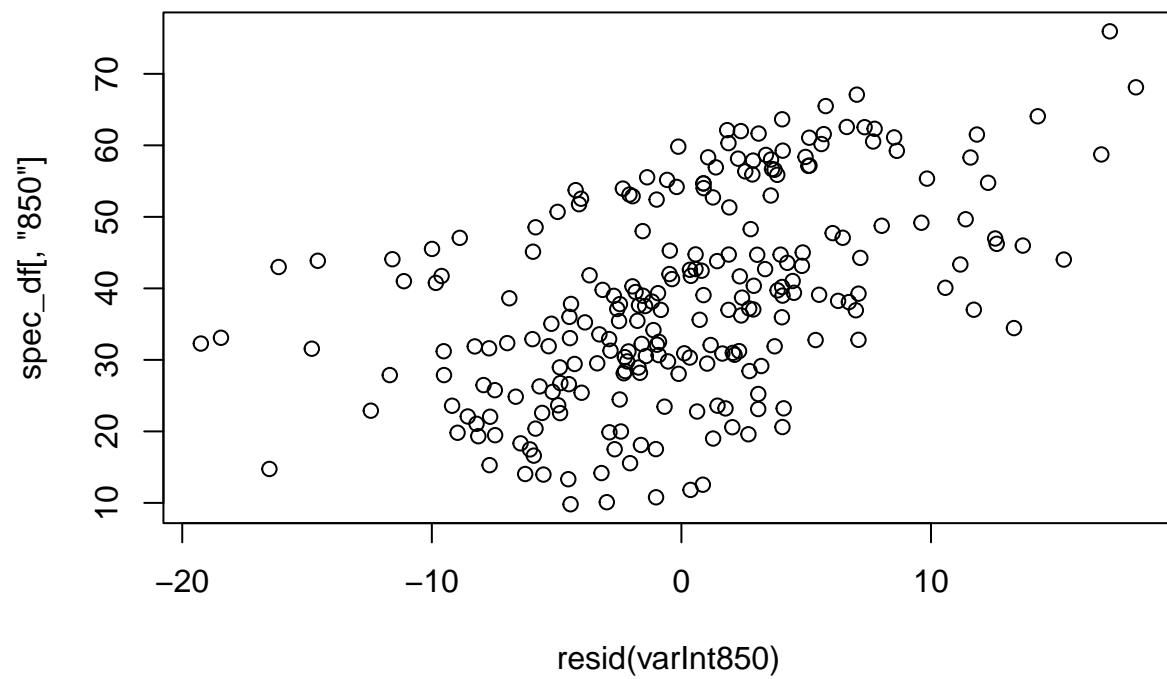
Variable intercept - fixed slope models

1. Linearity

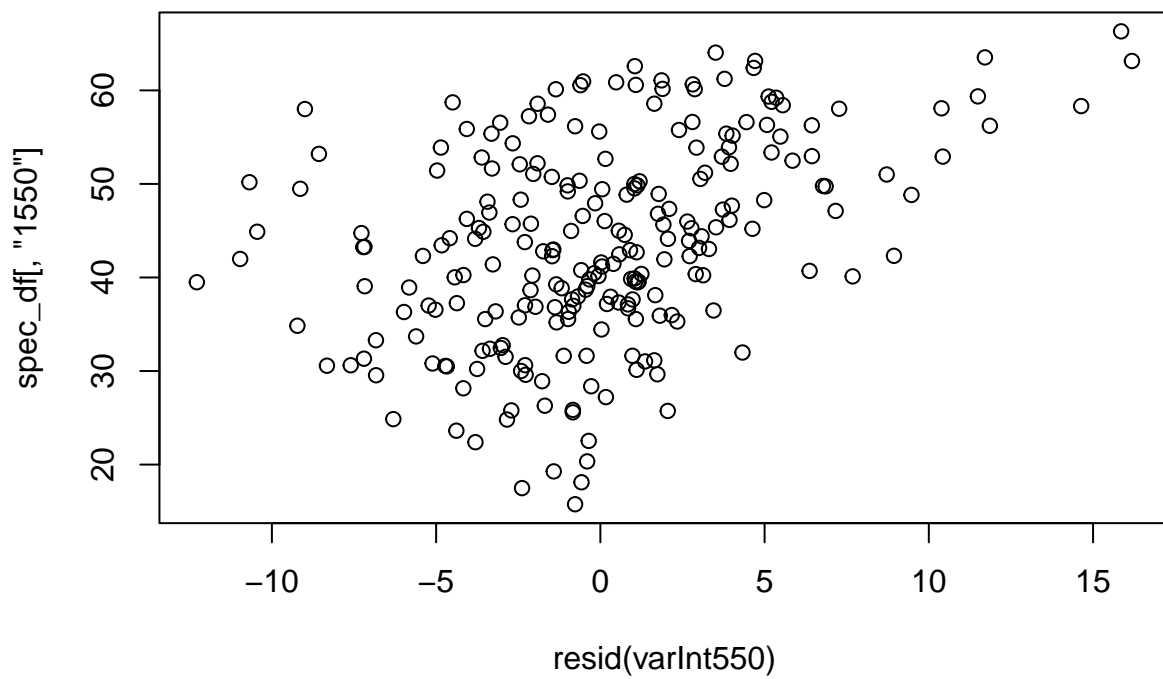
```
plot(resid(varInt550), spec_df[, '550'])
```



```
plot(resid(varInt850), spec_df[, '850'])
```

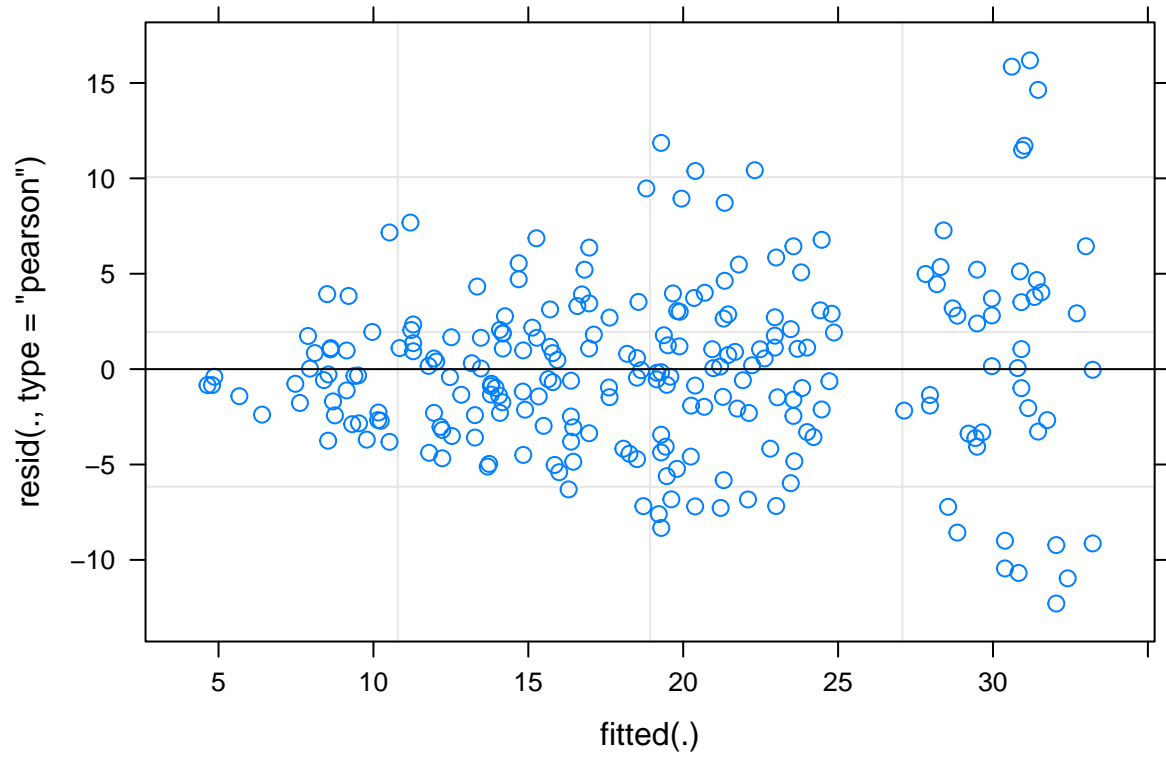
```
plot(resid(varInt550), spec_df[, '1550'])
```



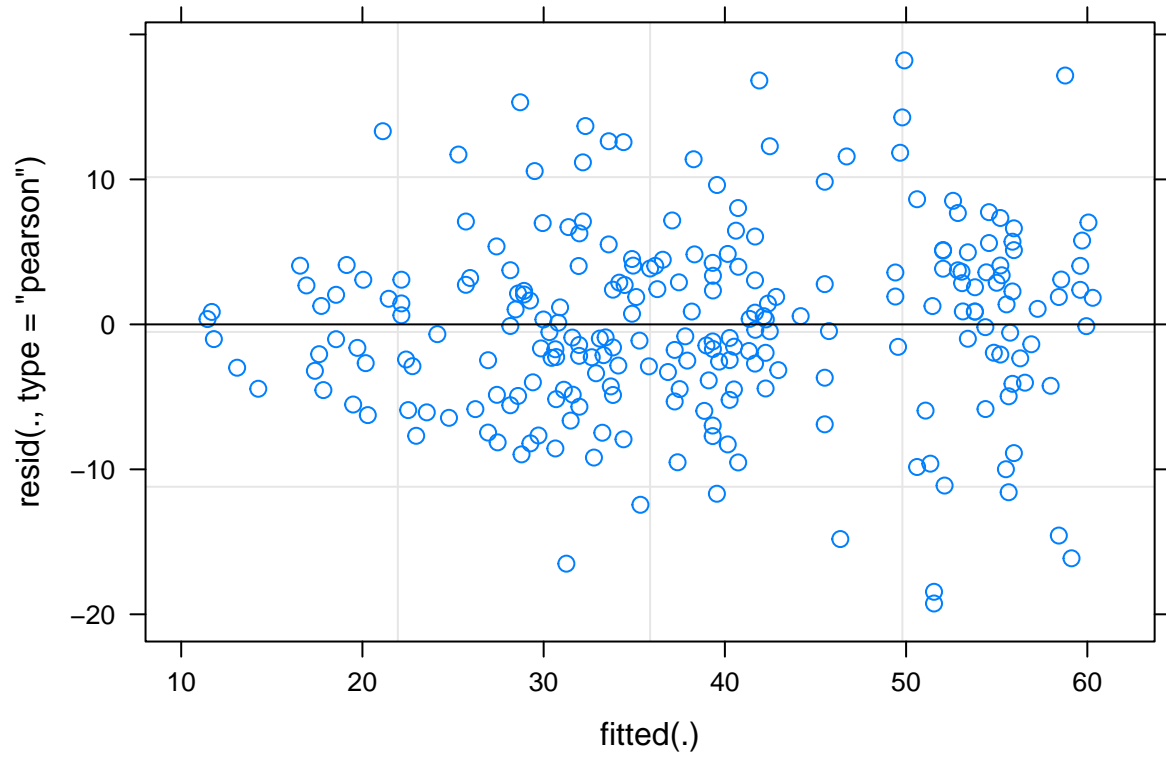
According to Michael Palmeri, a plot of random points indicates linearity. These plots look pretty random to me.

2. Homoscedasticity

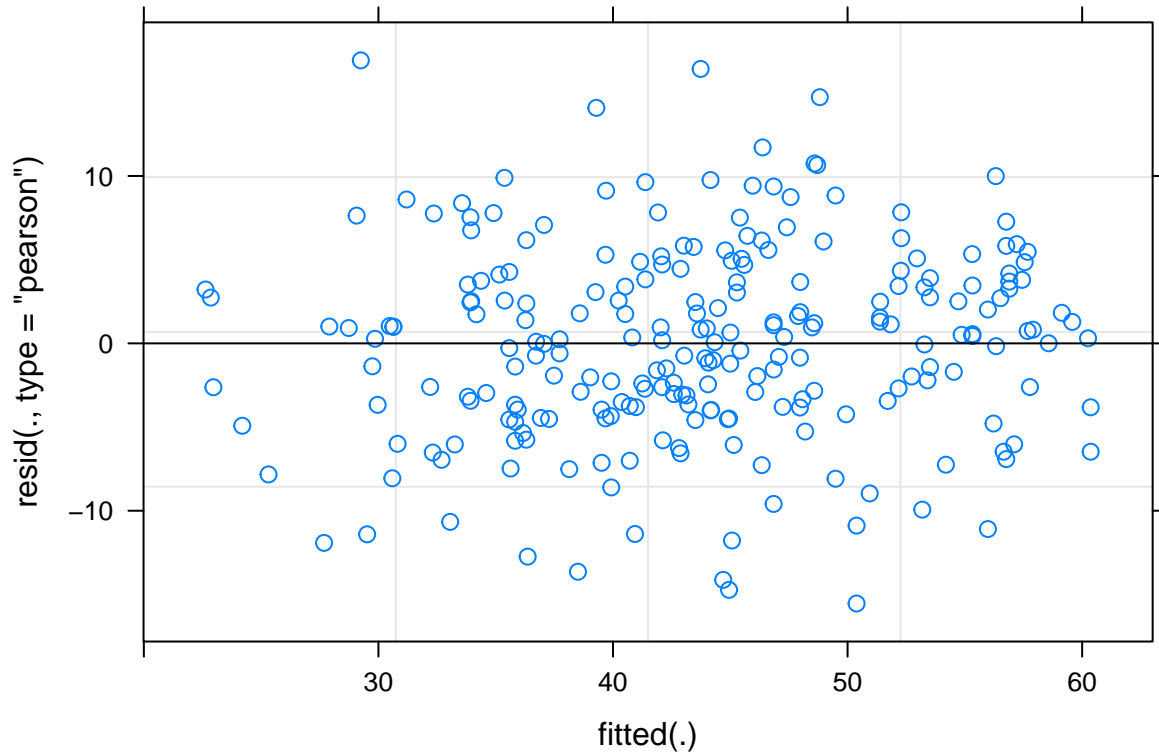
```
plot(varInt550)
```



```
plot(varInt850)
```



```
plot(varInt1550)
```

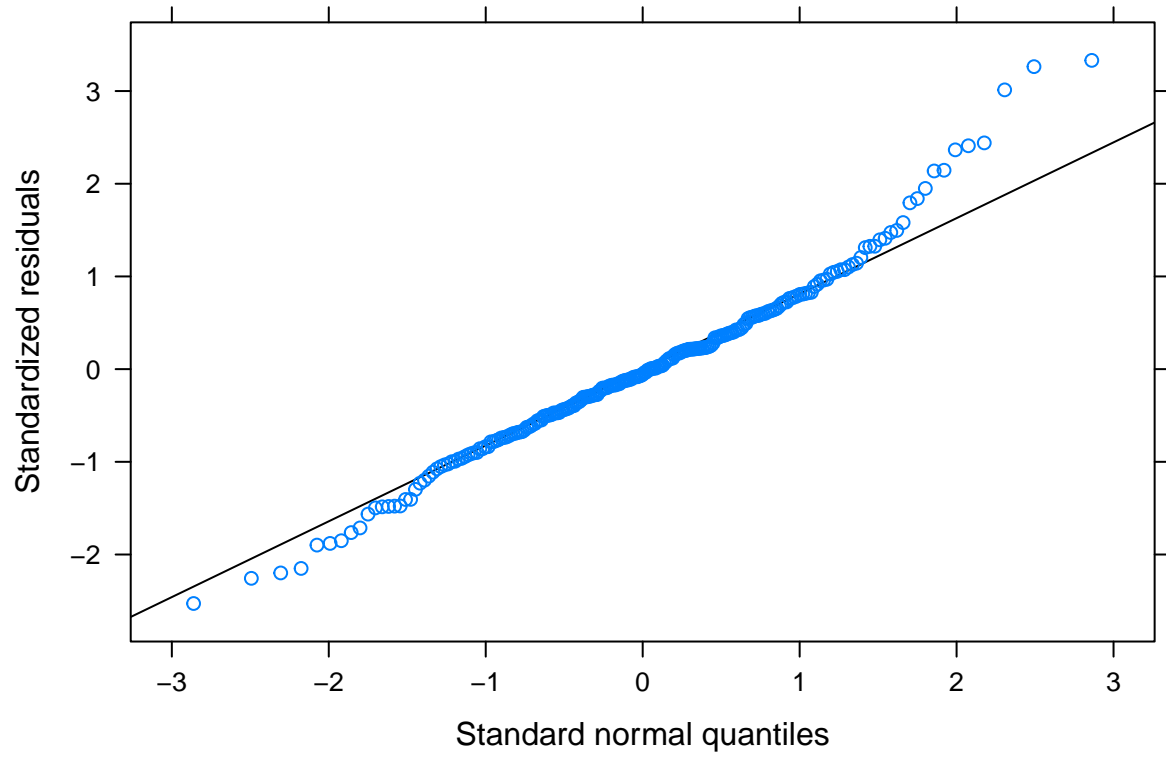


The 550 and 850 models appear to show heteroscedasticity, but the 1500 model looks pretty homoscedastic (sausage shaped).

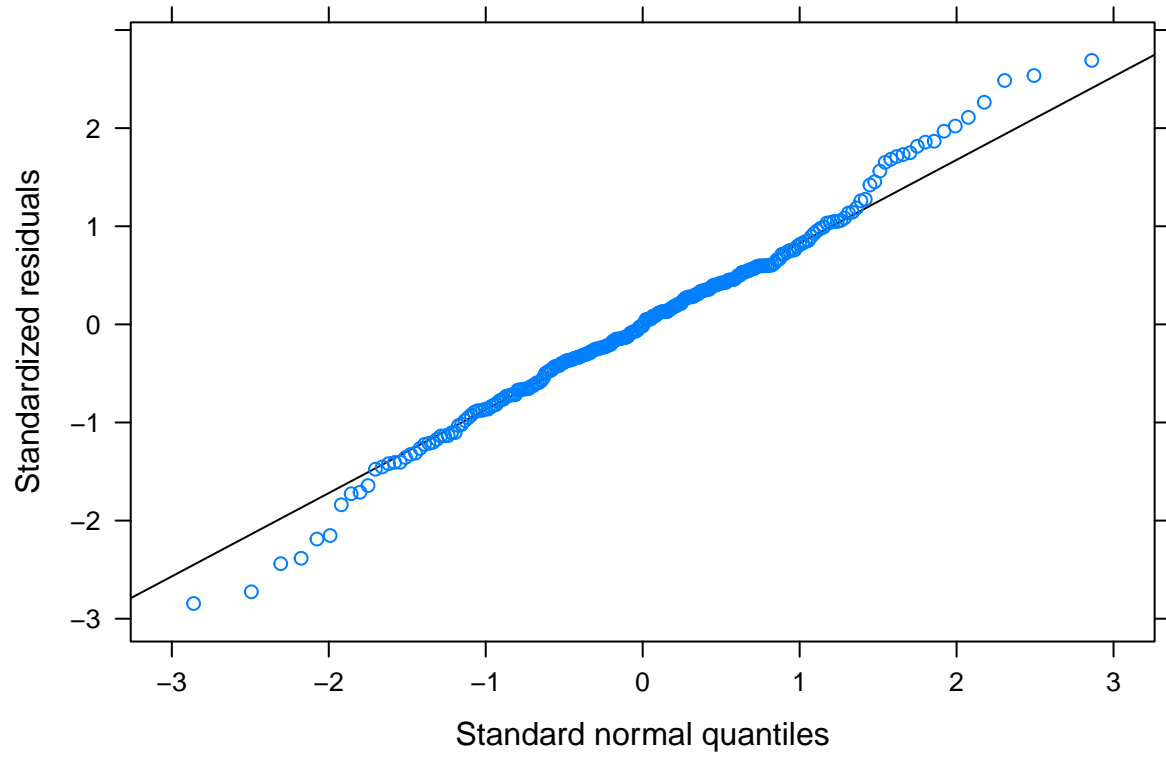
Question 4: Should I transform reflectance values? Some wavelengths show heteroscedasticity while others don't. Transforming could also make interpretation more difficult. Further, Schielzeth et al. (2020) demonstrate linear mixed models to be robust to heteroscedasticity.

3. Normally distributed residuals

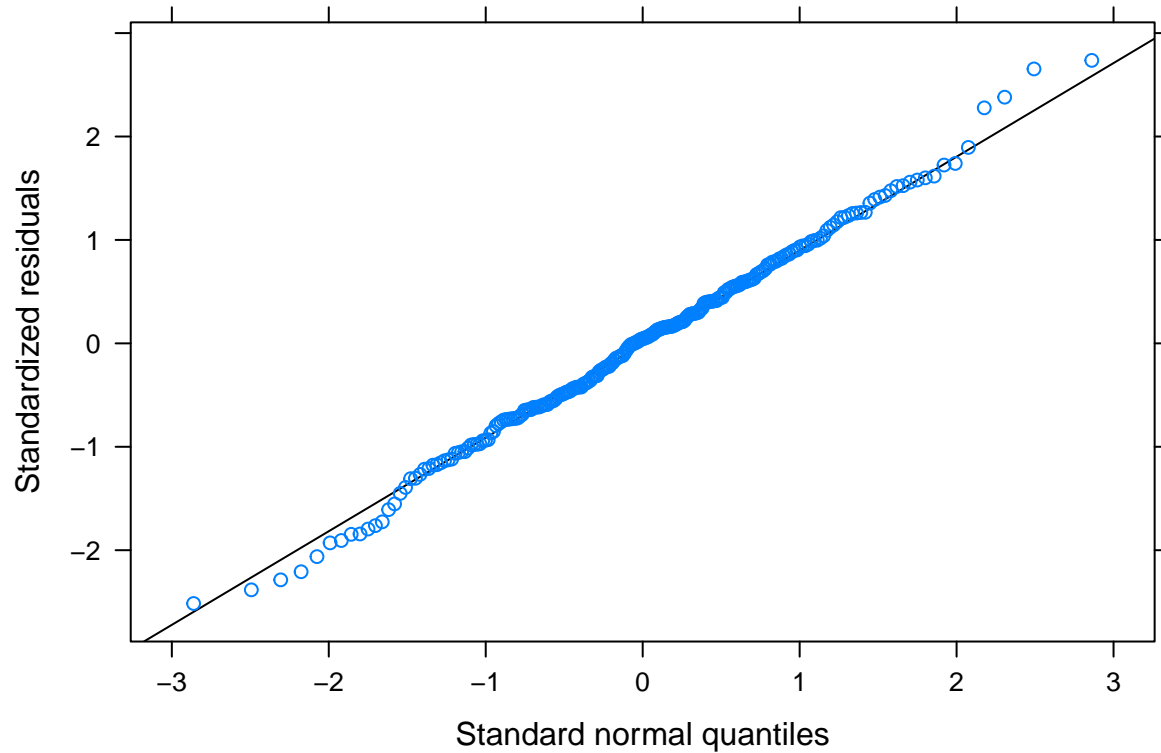
```
library(lattice)
qqmath(varInt550)
```



```
qqmath(varInt850)
```



```
qqmath(varInt1550)
```



Not the best, not the worst.

References

- Palmeri, M.(n.d.) Chapter 18: Testing the assumptions of multilevel models. https://ademos.people.uic.edu/Chapter18.html#1_preface
- Schielzeth, H., Dingemanse, N.J., Nakagawa, S., Westneat, D.F., Alaguela, H., Teplitsky, C., Reale, D., Dochtermann, N.A., Garamszegi, L.Z., Araya-Ajoy, Y. (2020). Robustness of linear mixed-effects models to violations of distributional assumptions. *Methods in Ecology and Evolution*, 11:1141-1152. <https://besjournals.onlinelibrary.wiley.com/doi/epdf/10.1111/2041-210X.13434>