```
# Negative Binomial models ----------------------------------------

#checking compatibility of glmmTMB and lme4 packages
packageVersion("glmmTMB")
```

```
## [1] '1.1.13'
```

```
packageVersion("lme4")
```

```
## [1] '1.1.37'
```

```
packageVersion("Matrix")
```

```
## [1] '1.7.4'
```

```
packageVersion("TMB")
```

```
## [1] '1.9.18'
```

```
packageVersion("mgcv")
```

```
## [1] '1.9.1'
```

```
sapply(c("Matrix", "TMB", "lme4", "glmmTMB"), find.package)
```

```
##                                                                 Matrix
##  "/Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/Matrix"
##                                                                    TMB
##     "/Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/TMB"
##                                                                   lme4
##    "/Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/lme4"
##                                                                 glmmTMB
## "/Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/library/glmmTMB"
```

```
#install.packages("remotes")
#remotes::install_version("lme4", version = "1.1.35")
#remotes::install_version("glmmTMB", version = "1.1.13")
#install.packages("Matrix", type="source")
#install.packages("TMB", type="source")
#install.packages("glmmTMB", type="source", INSTALL_opts = "--no-multiarch --config
ure-vars='INCLUDE_DIRS=/Library/Frameworks/R.framework/Versions/4.4-arm64/Resource
s/library/TMB/include'")

#simple test of glmmTMB - failing due to incompatibility.  Need older source
#of lme4, but not possible with this version of MacOS.
#library(glmmTMB)
#test_data <- data.frame(
 # y = rpois(100, 10),
#  x = runif(100),
 # g = rep(letters[1:10], each = 10)
#)

#m_test <- glmmTMB(y ~ x + (1|g), family = nbinom2, data = test_data)
#summary(m_test)



#need to convert to 'factor' for negative binomial
stand_ID_filtered$patch_name <- as.factor(stand_ID_filtered$patch_name)
stand_ID_filtered$stand_ID <- as.factor(stand_ID_filtered$stand_ID)

library(mgcv)
```

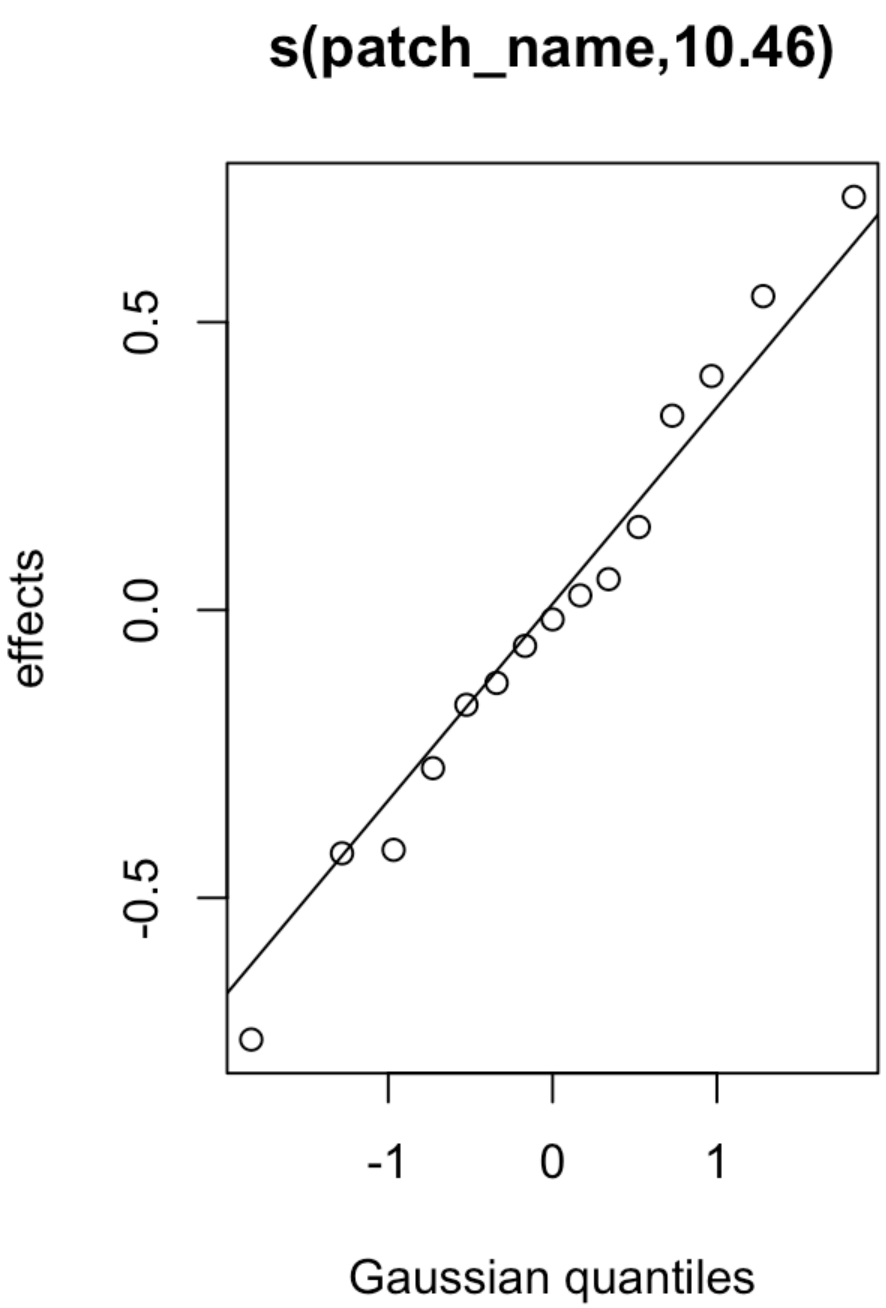
```
## Loading required package: nlme
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:lme4':
##
##     lmList
##
## The following object is masked from 'package:dplyr':
##
##     collapse
##
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
##
## Attaching package: 'mgcv'
##
## The following objects are masked from 'package:brms':
##
##     s, t2
```

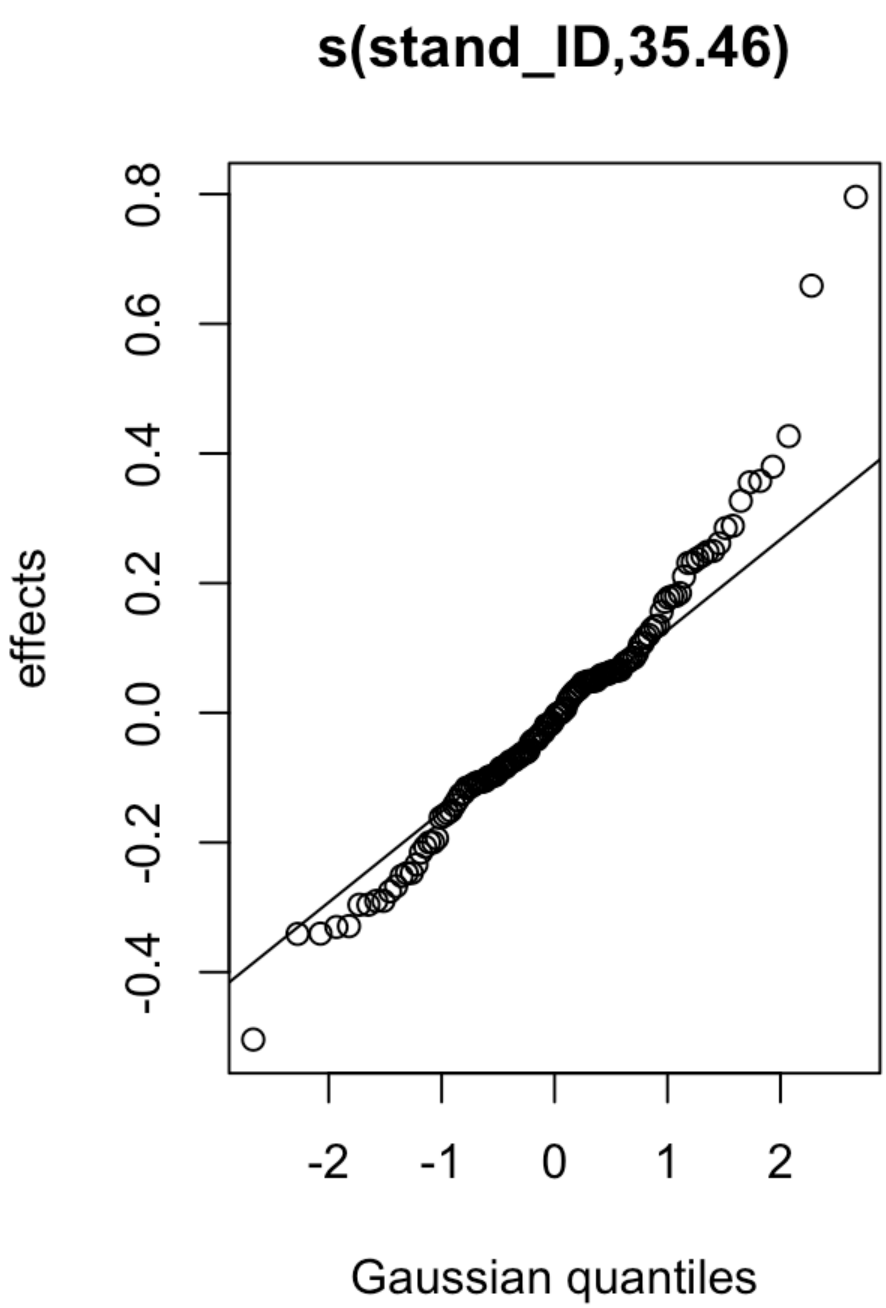
```
##NB with Oak

Oak_nb_model <- gam(round(clean_complete) ~ Percent_Oak +
                      s(patch_name, bs = "re") +  # random effect for patch_name
                      s(stand_ID, bs = "re"),     # random effect for stand_ID
                    family = nb(),                # negative binomial
                    method = "REML",
                    data = stand_ID_filtered)
summary(Oak_nb_model)
```

```
##
## Family: Negative Binomial(2.488)
## Link function: log
##
## Formula:
## round(clean_complete) ~ Percent_Oak + s(patch_name, bs = "re") +
##     s(stand_ID, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.5213     0.1636  21.524  < 2e-16 ***
## Percent_Oak   0.6857     0.2653   2.585  0.00975 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df Chi.sq p-value
## s(patch_name) 10.46     14 185.75 < 2e-16 ***
## s(stand_ID)   35.46    129  65.87 0.00472 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.456   Deviance explained = 56.1%
## -REML =  942.8  Scale est. = 1         n = 190
```

```
plot(Oak_nb_model, pages = 1)
```


s(patch_name,10.46)
effects
0.5
0.0
-0.5
-1
0
1
Gaussian quantiles

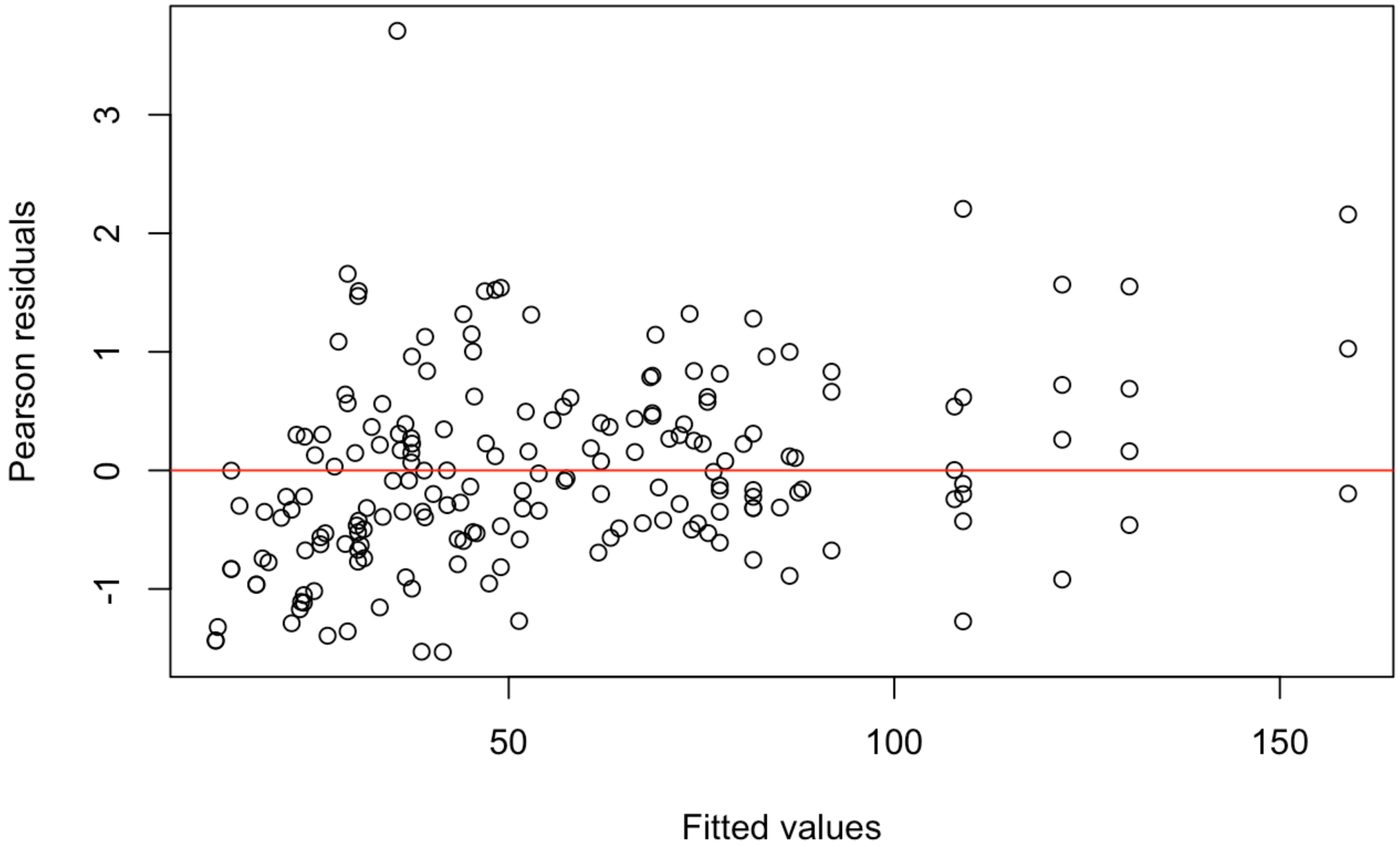

s(stand_ID,35.46)
effects
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-2
-1
0
1
2
Gaussian quantiles

```
# Fitted values
fitted_vals_oak <- fitted(Oak_nb_model)

# Pearson residuals
pearson_resid_oak <- residuals(Oak_nb_model, type = "pearson")

# Residual degrees of freedom
rdf_oak <- df.residual(Oak_nb_model)

plot(fitted_vals_oak, pearson_resid_oak,
     xlab="Fitted values", ylab="Pearson residuals")
abline(h=0, col="red")
```


Pearson residuals
3
2
1
0
-1
50
100
150
Fitted values

```
# Dispersion ratio
dispersion_oak <- sum(pearson_resid_oak^2) / rdf_oak
dispersion_oak
```

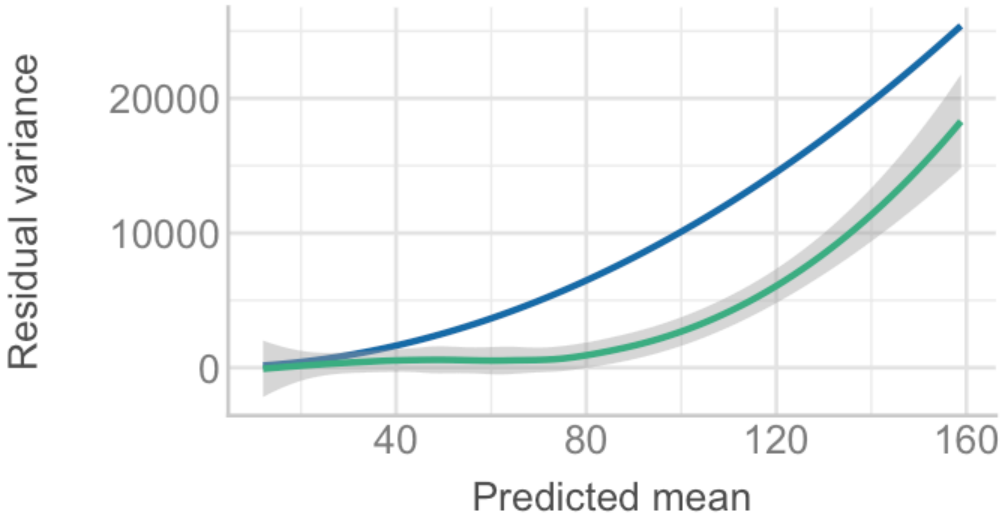
```
## [1] 0.8515649
```

```
#dispersion = 0.852, indicating that there is no over (or much under) dispersion

performance::check_model(Oak_nb_model, residual_type = "normal")
```

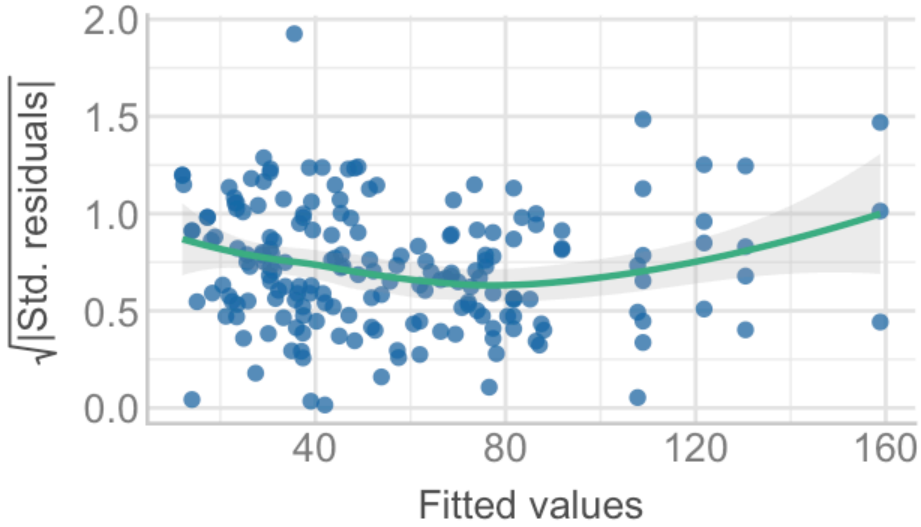
Misspecified dispersion and zero-inflation

Observed residual variance (green) should follow predicted residual variance (blue)



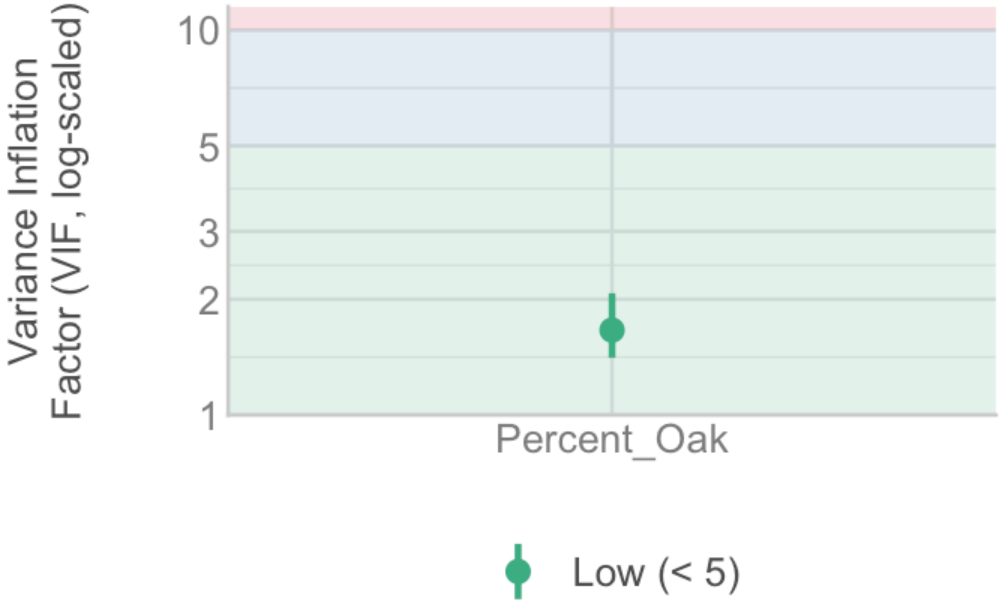
Homogeneity of Variance

Residual variance should be flat and horizontal



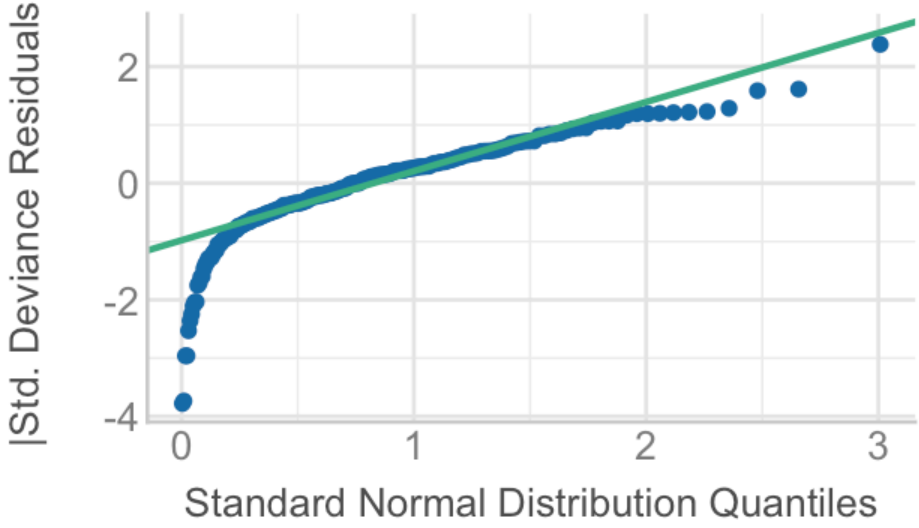
Collinearity

High collinearity (VIF) may inflate parameter uncertainty



Normality of Residuals

Dots should fall along the line

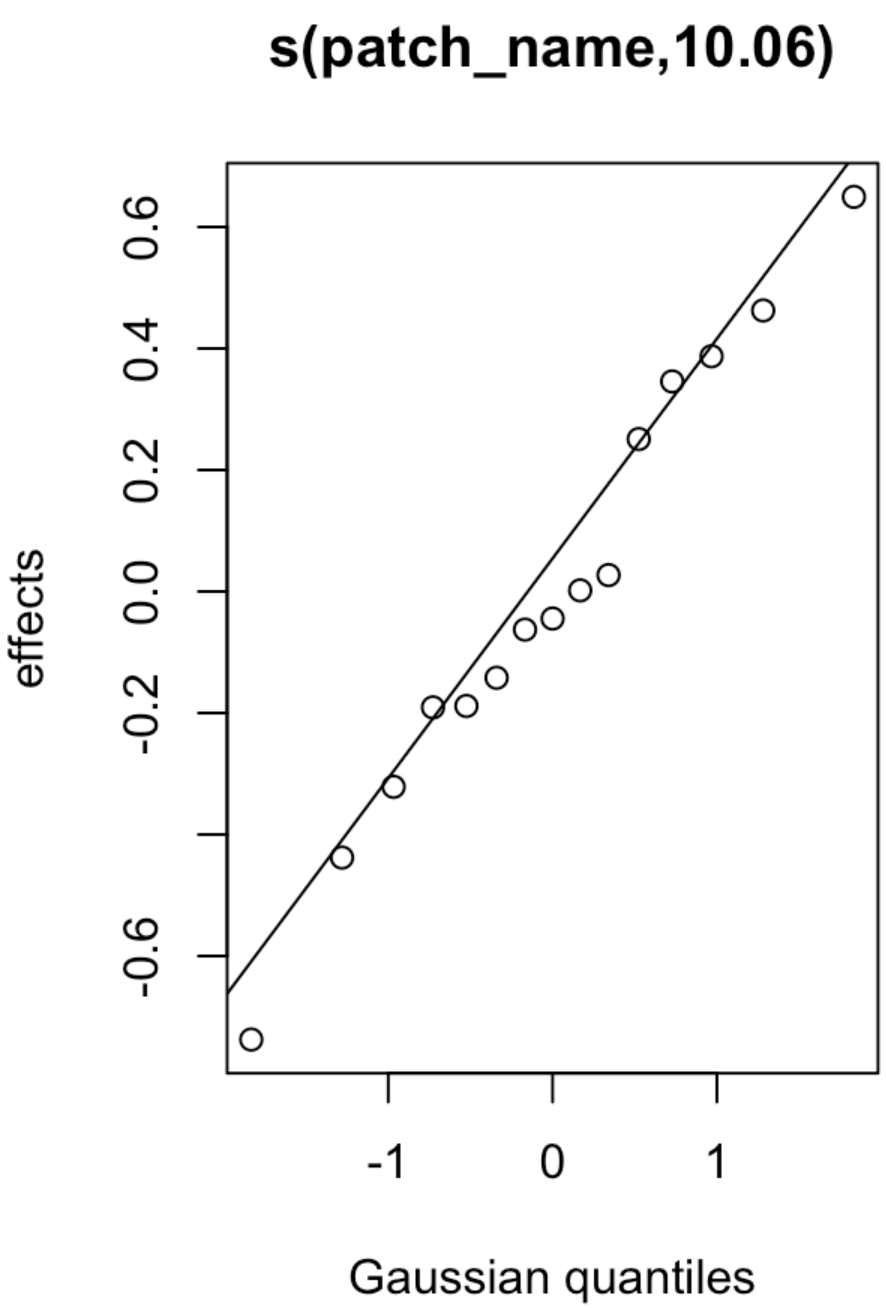


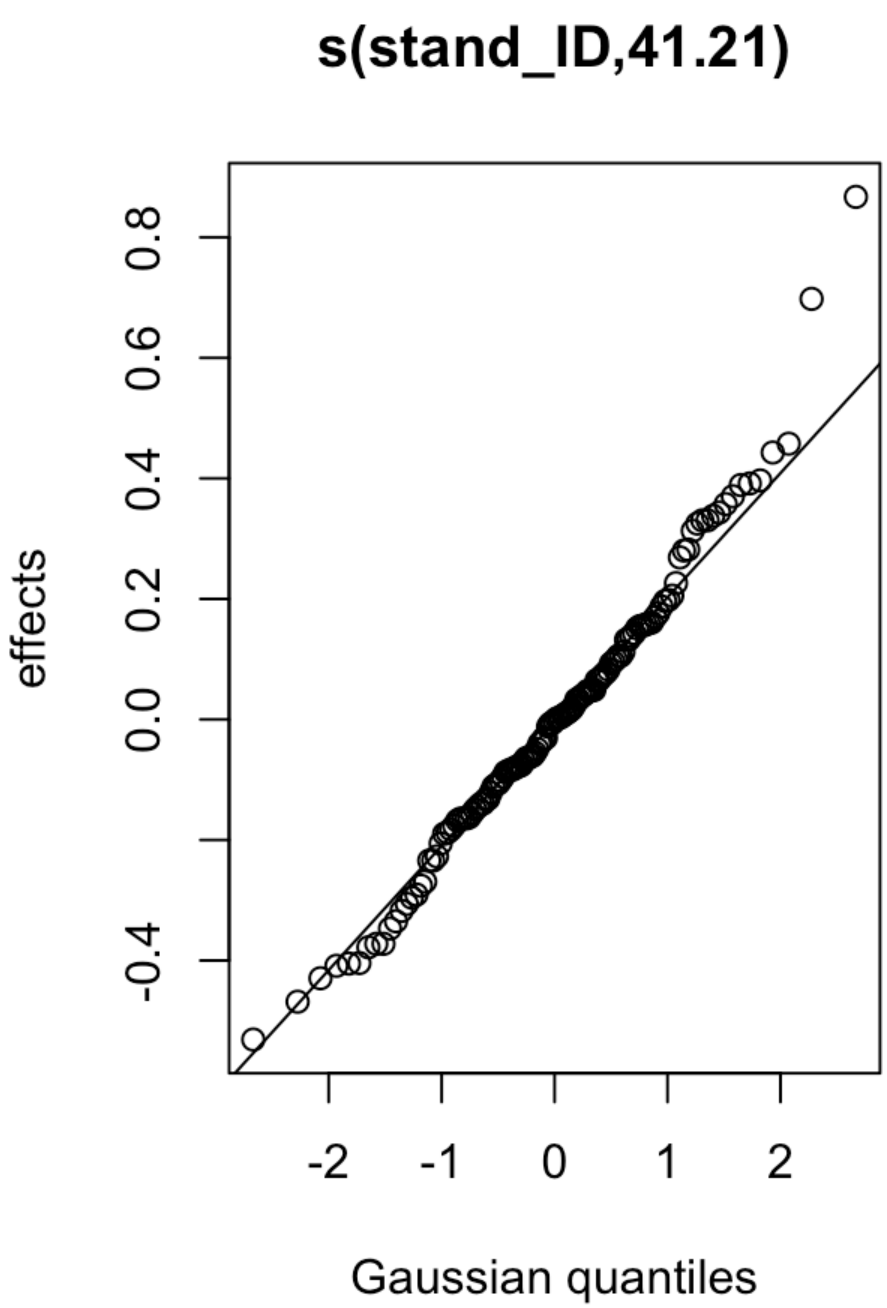
```
##NB with Pine

Pine_nb_model <- gam(round(clean_complete) ~ Percent_Pine +
                       s(patch_name, bs = "re") +  # random effect for patch_name
                       s(stand_ID, bs = "re"),     # random effect for stand_ID
                     family = nb(),                # negative binomial
                     method = "REML",
                     data = stand_ID_filtered)
summary(Pine_nb_model)
```

```
##
## Family: Negative Binomial(2.518)
## Link function: log
##
## Formula:
## round(clean_complete) ~ Percent_Pine + s(patch_name, bs = "re") +
##     s(stand_ID, bs = "re")
##
## Parametric coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.7206     0.1376  27.036   <2e-16 ***
## Percent_Pine   0.2016     0.3034   0.664    0.506
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df Chi.sq p-value
## s(patch_name) 10.06     14 161.96 < 2e-16 ***
## s(stand_ID)   41.21    129  82.84 0.00112 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.448   Deviance explained = 57.9%
## -REML = 945.55  Scale est. = 1         n = 190
```

```
plot(Pine_nb_model, pages = 1)
```


s(patch_name,10.06)
effects
-0.6
-0.2
0.0
0.2
0.4
0.6
-1
0
1
Gaussian quantiles

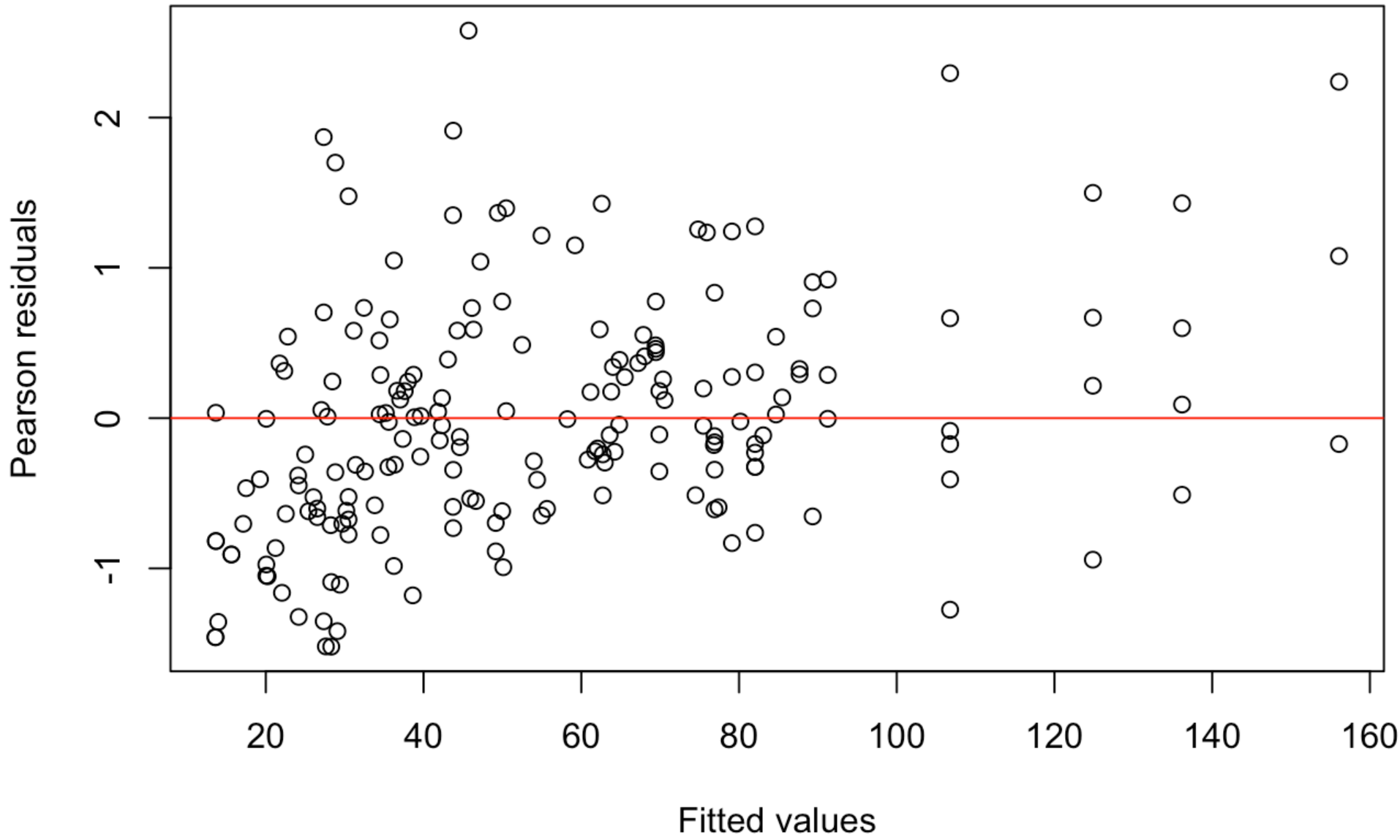

s(stand_ID,41.21)
effects
-0.4
0.0
0.2
0.4
0.6
0.8
-2
-1
0
1
2
Gaussian quantiles

```r
# Fitted values
fitted_vals_pine <- fitted(Pine_nb_model)

# Pearson residuals
pearson_resid_pine <- residuals(Pine_nb_model, type = "pearson")

# Residual degrees of freedom
rdf_pine <- df.residual(Pine_nb_model)

plot(fitted_vals_pine, pearson_resid_pine,
     xlab="Fitted values", ylab="Pearson residuals")
abline(h=0, col="red")
```


Pearson residuals
2
1
0
-1
20
40
60
80
100
120
140
160
Fitted values

```
# Dispersion ratio
dispersion_pine <- sum(pearson_resid_pine^2) / rdf_pine
dispersion_pine
```

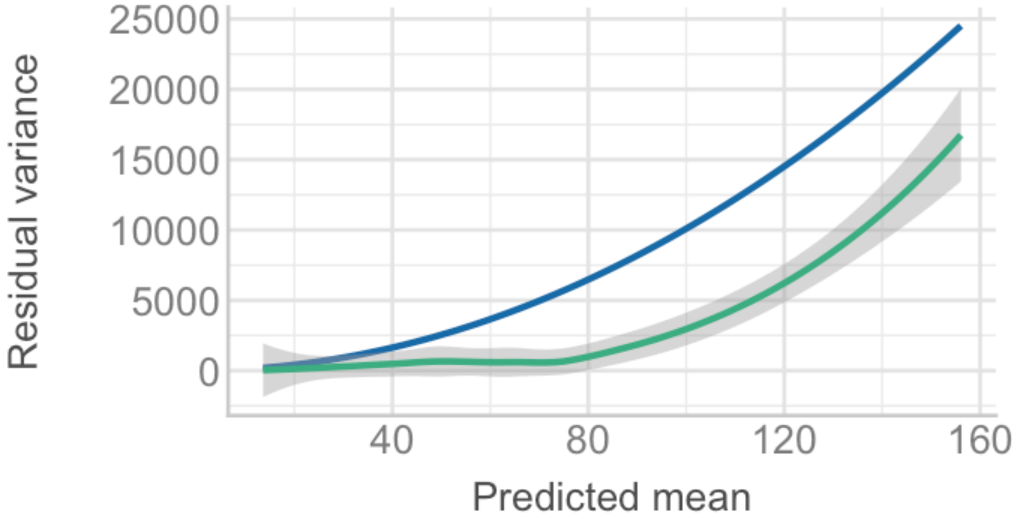
```
## [1] 0.8448824
```

```
#dispersion = 0.845, indicating that there is no over (or much under) dispersion

performance::check_model(Pine_nb_model, residual_type = "normal")
```

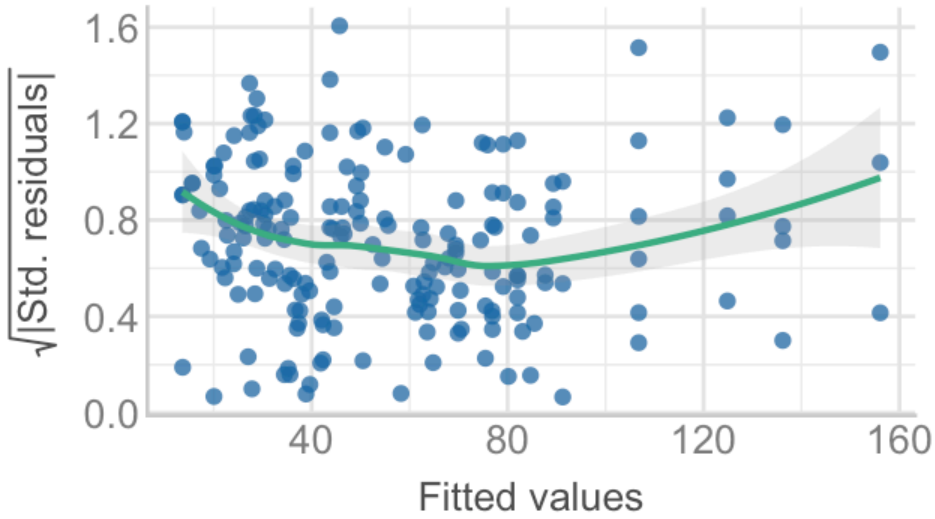
Misspecified dispersion and zero-inflation

Observed residual variance (green) should follow predicted residual variance (blue)



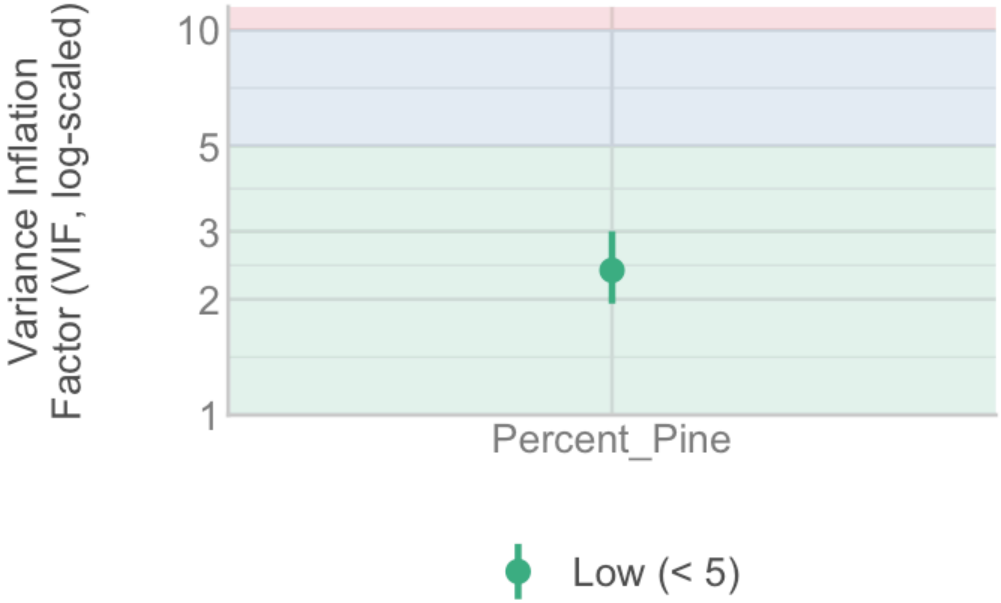
Homogeneity of Variance

Residual variance should be flat and horizontal



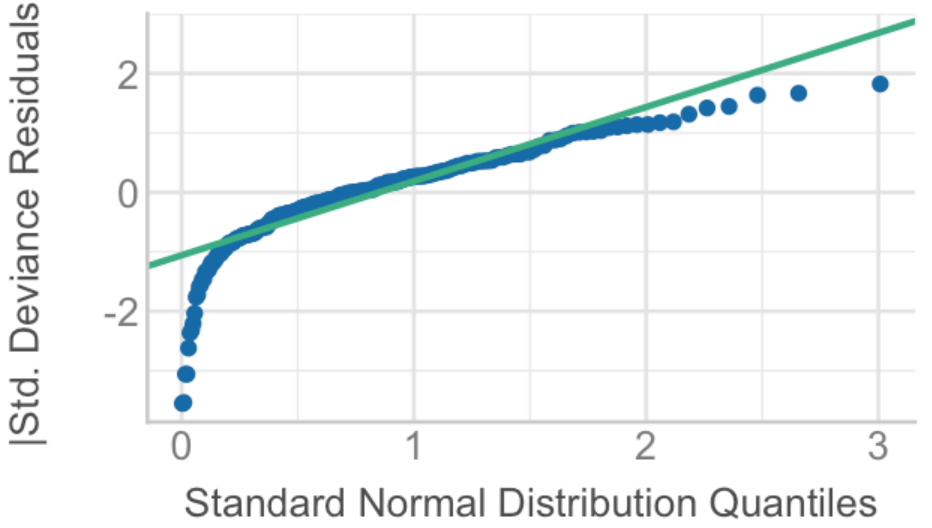
Collinearity

High collinearity (VIF) may inflate parameter uncertainty



Normality of Residuals

Dots should fall along the line

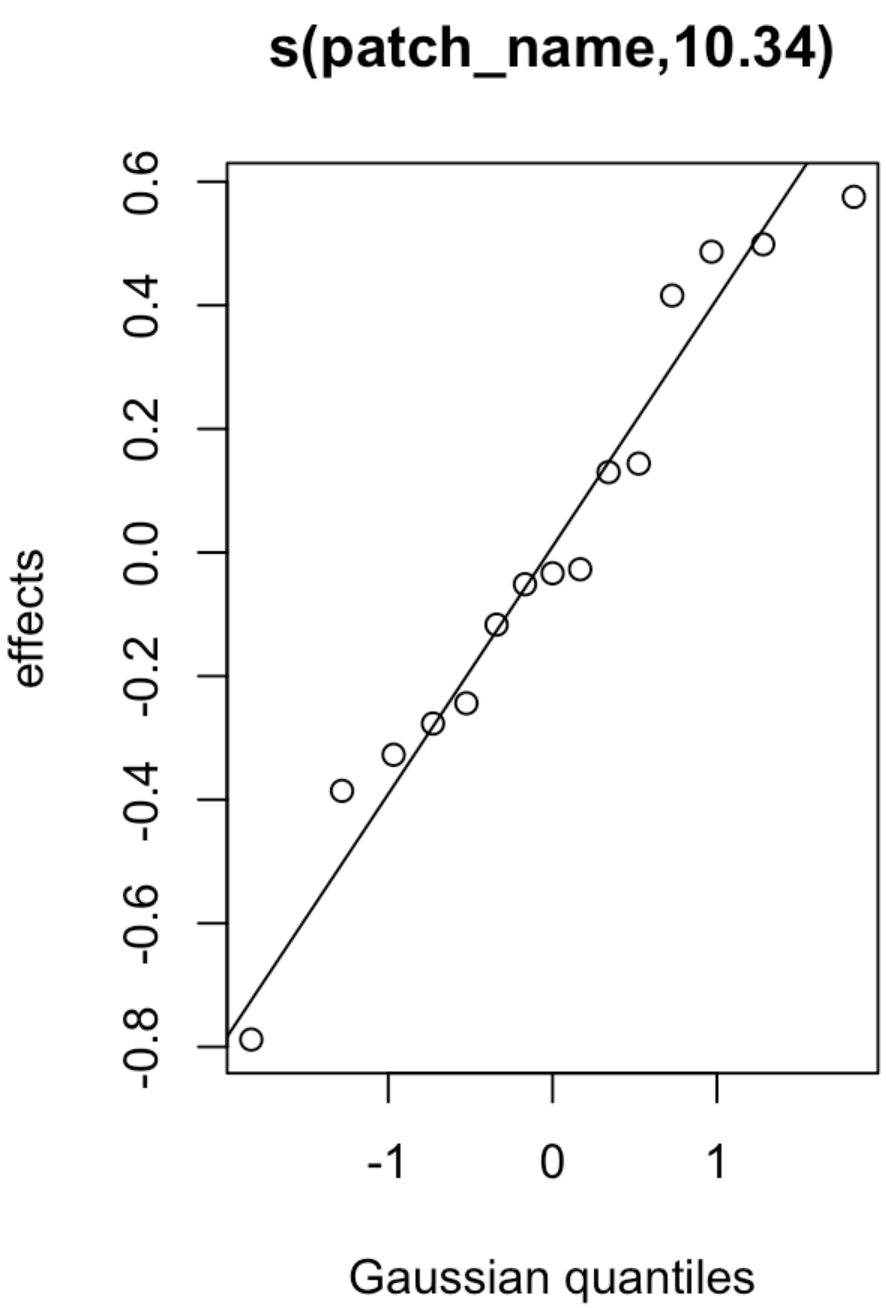


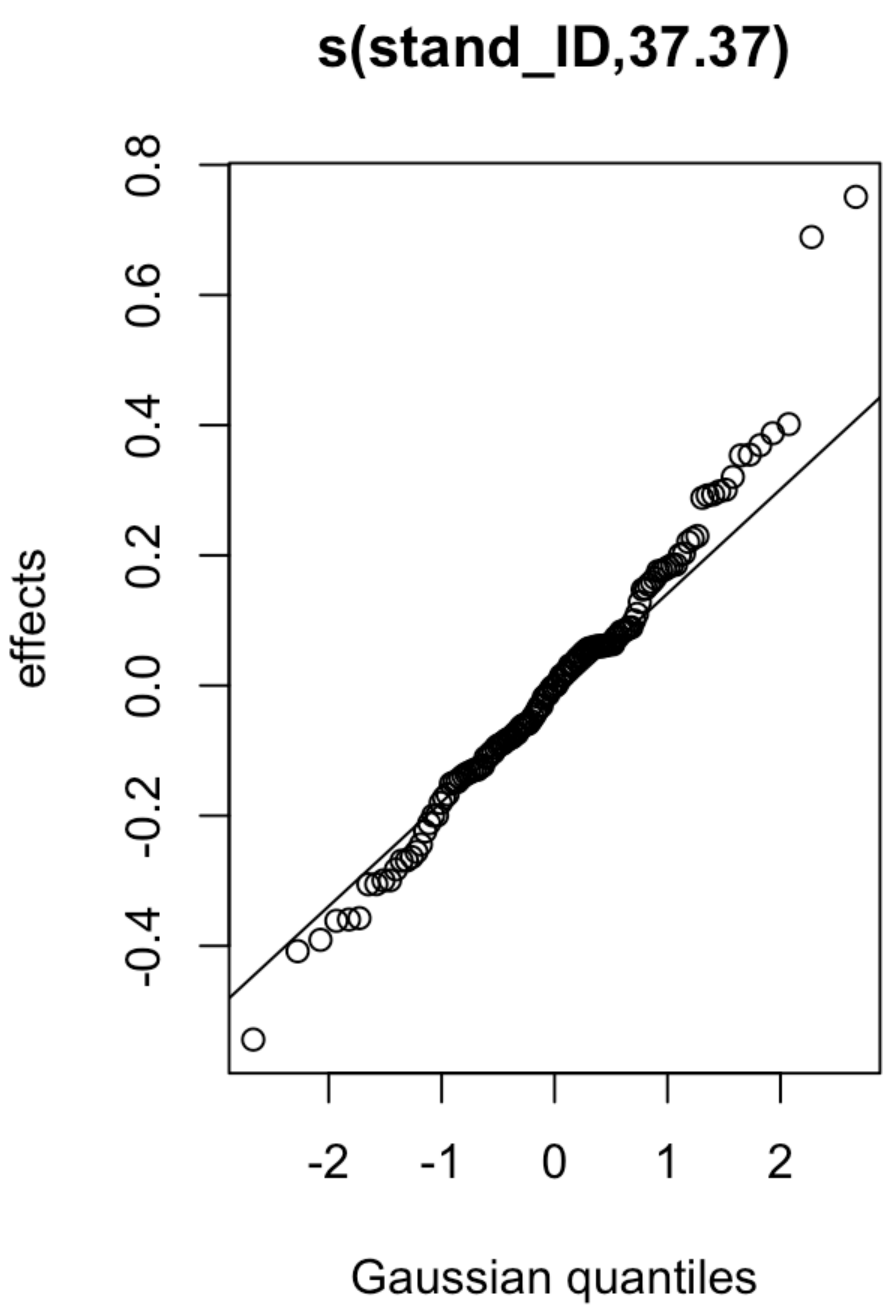
```
##NB with both Oak and Pine

Both_nb_model <- gam(round(clean_complete) ~ Percent_Oak + Percent_Pine +
                       s(patch_name, bs = "re") +  # random effect for patch_name
                       s(stand_ID, bs = "re"),     # random effect for stand_ID
                     family = nb(),                # negative binomial
                     method = "REML",
                     data = stand_ID_filtered)
summary(Both_nb_model)
```

```
##
## Family: Negative Binomial(2.593)
## Link function: log
##
## Formula:
## round(clean_complete) ~ Percent_Oak + Percent_Pine + s(patch_name,
##     bs = "re") + s(stand_ID, bs = "re")
##
## Parametric coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.2787     0.1872  17.511  < 2e-16 ***
## Percent_Oak    1.0766     0.3048   3.532 0.000412 ***
## Percent_Pine   0.8024     0.3368   2.382 0.017202 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                 edf Ref.df Chi.sq p-value
## s(patch_name) 10.34     14 162.90 < 2e-16 ***
## s(stand_ID)   37.37    128  70.22 0.00175 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.455   Deviance explained = 58.5%
## -REML = 940.41  Scale est. = 1         n = 190
```

```
plot(Both_nb_model, pages = 1)
```


s(patch_name,10.34)
effects
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
-1
0
1
Gaussian quantiles

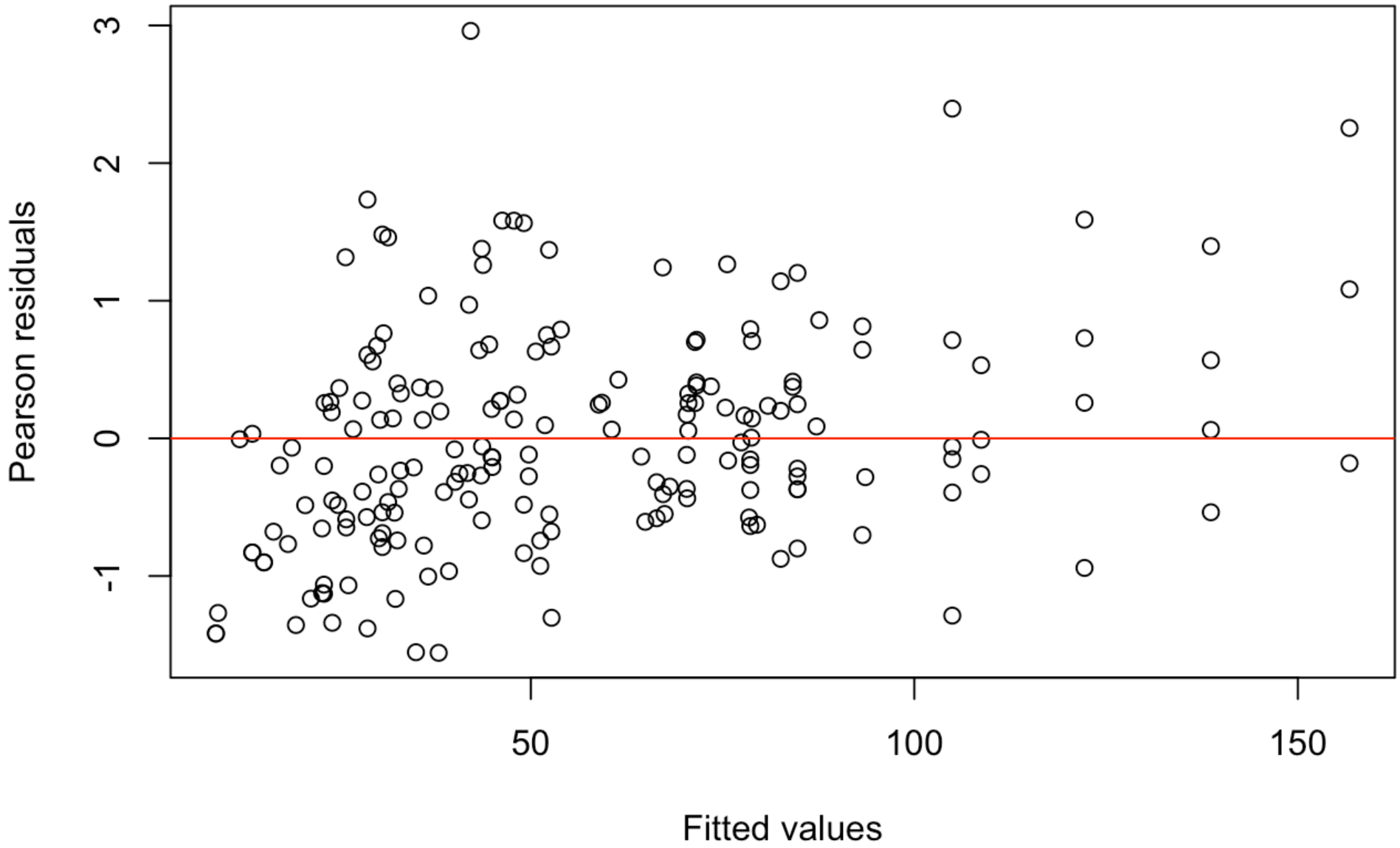

s(stand_ID,37.37)
effects
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
-2
-1
0
1
2
Gaussian quantiles

```r
# Fitted values
fitted_vals_both <- fitted(Both_nb_model)

# Pearson residuals
pearson_resid_both <- residuals(Both_nb_model, type = "pearson")

# Residual degrees of freedom
rdf_both <- df.residual(Both_nb_model)

plot(fitted_vals_both, pearson_resid_both,
     xlab="Fitted values", ylab="Pearson residuals")
abline(h=0, col="red")
```


Pearson residuals
3
2
1
0
-1
50
100
150
Fitted values

```r
# Dispersion ratio
dispersion_both <- sum(pearson_resid_both^2) / rdf_both
dispersion_both
```

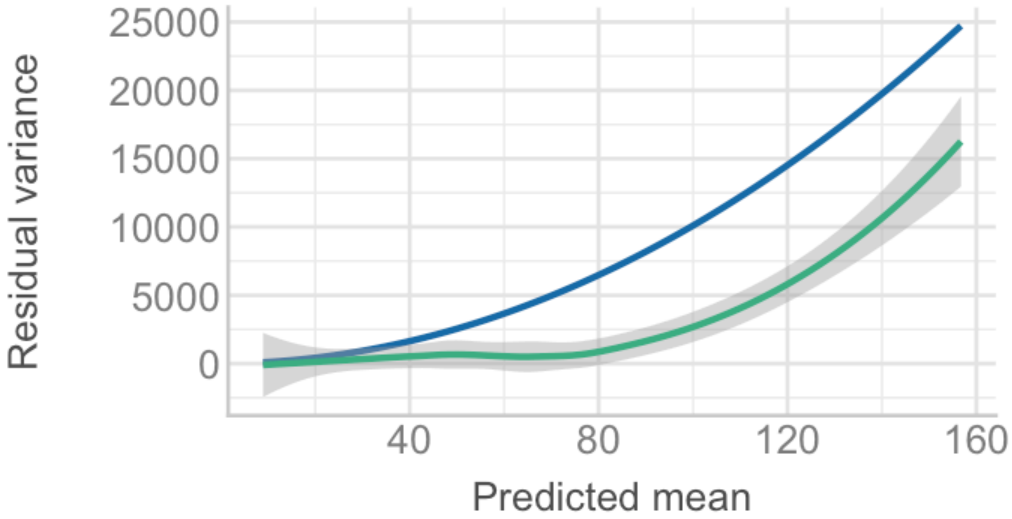
```
## [1] 0.851848
```

```r
#dispersion = 0.852, indicating that there is no over (or much under) dispersion

performance::check_model(Both_nb_model, residual_type = "normal")
```

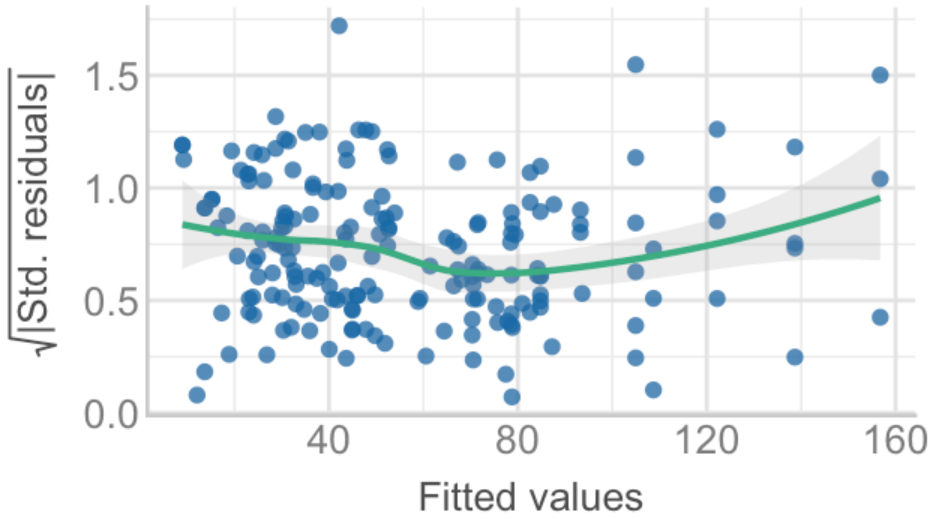
Misspecified dispersion and zero-inflation

Observed residual variance (green) should follow predicted residual variance (blue)



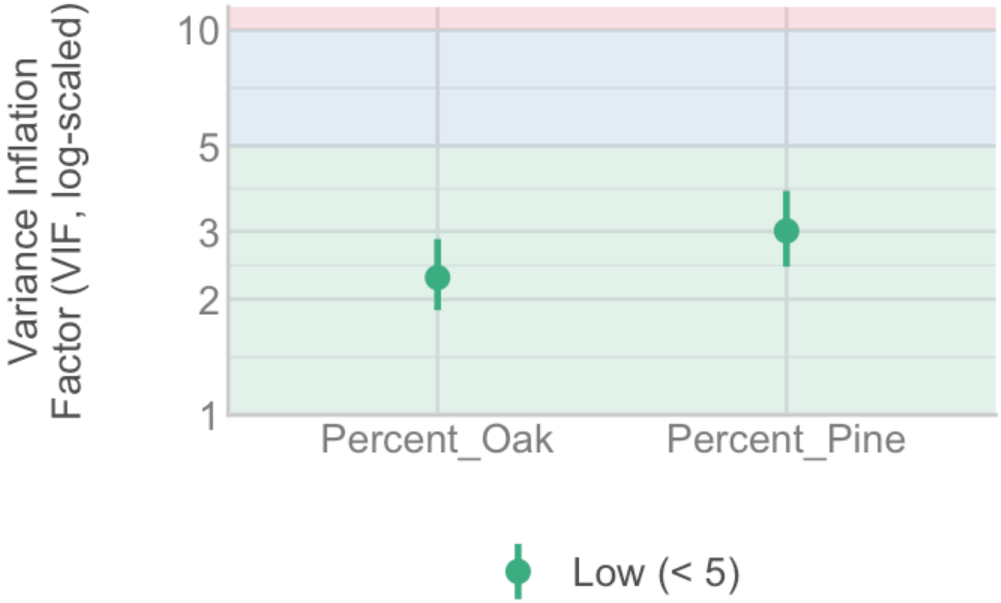
Homogeneity of Variance

Residual variance should be flat and horizontal



Collinearity

High collinearity (VIF) may inflate parameter uncertainty



Normality of Residuals

Dots should fall along the line