

Robust analysis

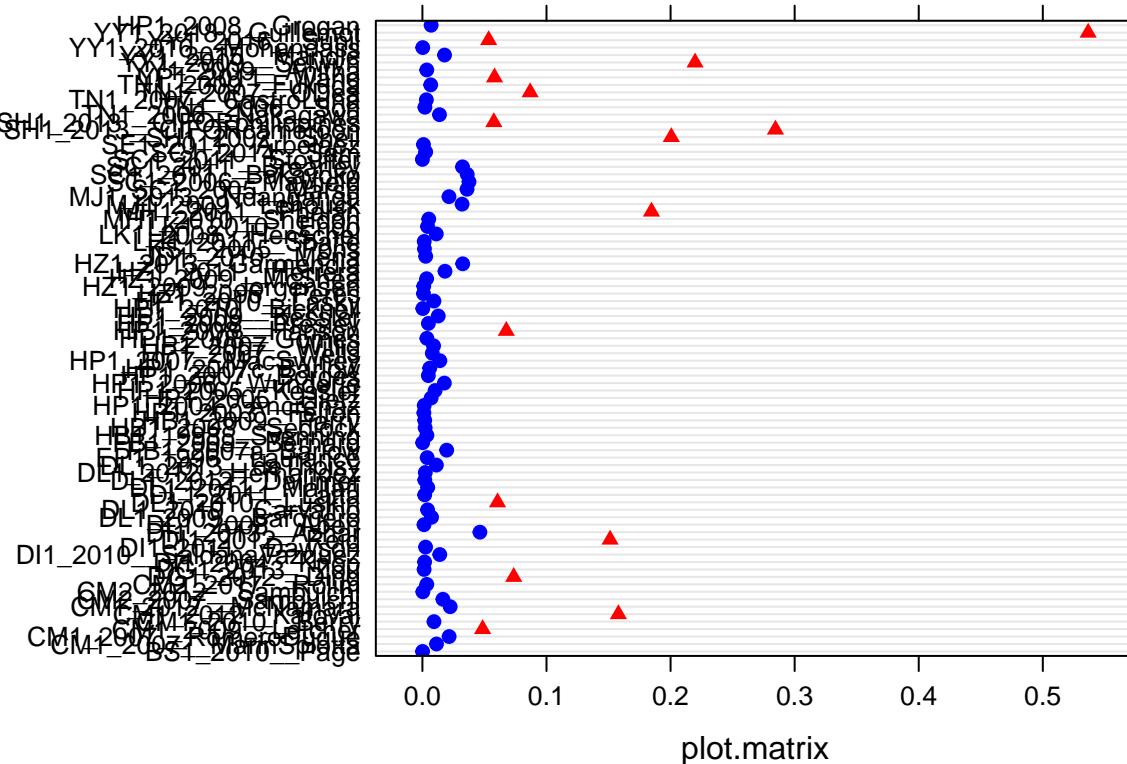
Total Abundance Model

I used the influence.ME package to identify influential Source IDs. The first measure I assessed was the Cook's distance. The cut off for this measure is: $4/n$, being n the number of groups, in this case 85. Cut off= 0.047

```
# Count number of source IDs that are greater than the cut off
length(cook_sources_abundance$Source_ID)

## [1] 14

# Plot the results
influence.ME::plot.estex(x=INFM1_abundance, which = "cook", cutoff = 0.047)
```



The second measure I assessed was the DFBETAS. The cut off for this measure is: $2/\sqrt{n}$, being n the number of groups, in this case 85. Cut off= 0.217

```
# Count number of source IDs that are greater than the cut off for any parameter
length(dfbetas_sources_abundance$Source_ID)
```

```
## [1] 54
```

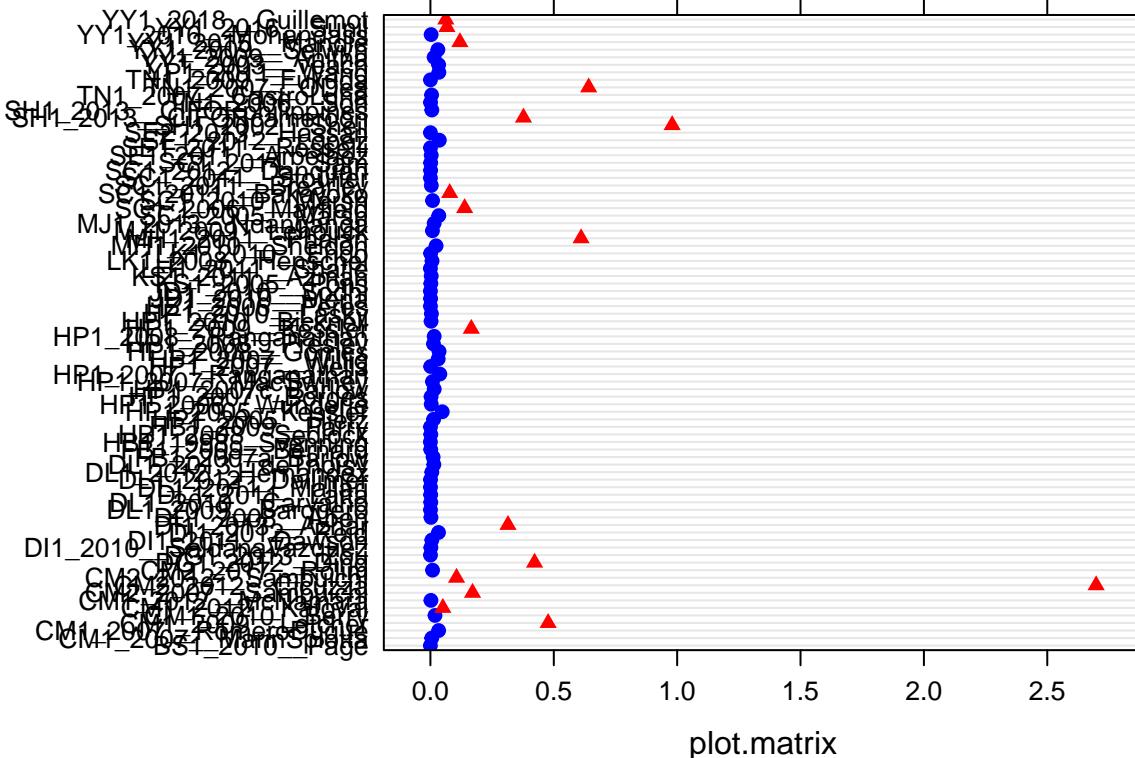
Species Richness model

Cut off Cook's distance = $4/84 = 0.048$

```
# Count number of source IDs that are greater than the cut off
length(cook_sources_richness$Source_ID)
```

```
## [1] 17
```

```
# Plot the results
influence.ME::plot.estex(x=INFM1_richness, which = "cook", cutoff = 0.05)
```



Cut off DFBETAS = $2/\sqrt{84} = 0.22$

```
# Count number of source IDs that are greater than the cut off
length(dfbetas_sources_richness$Source_ID)
```

```
## [1] 40
```

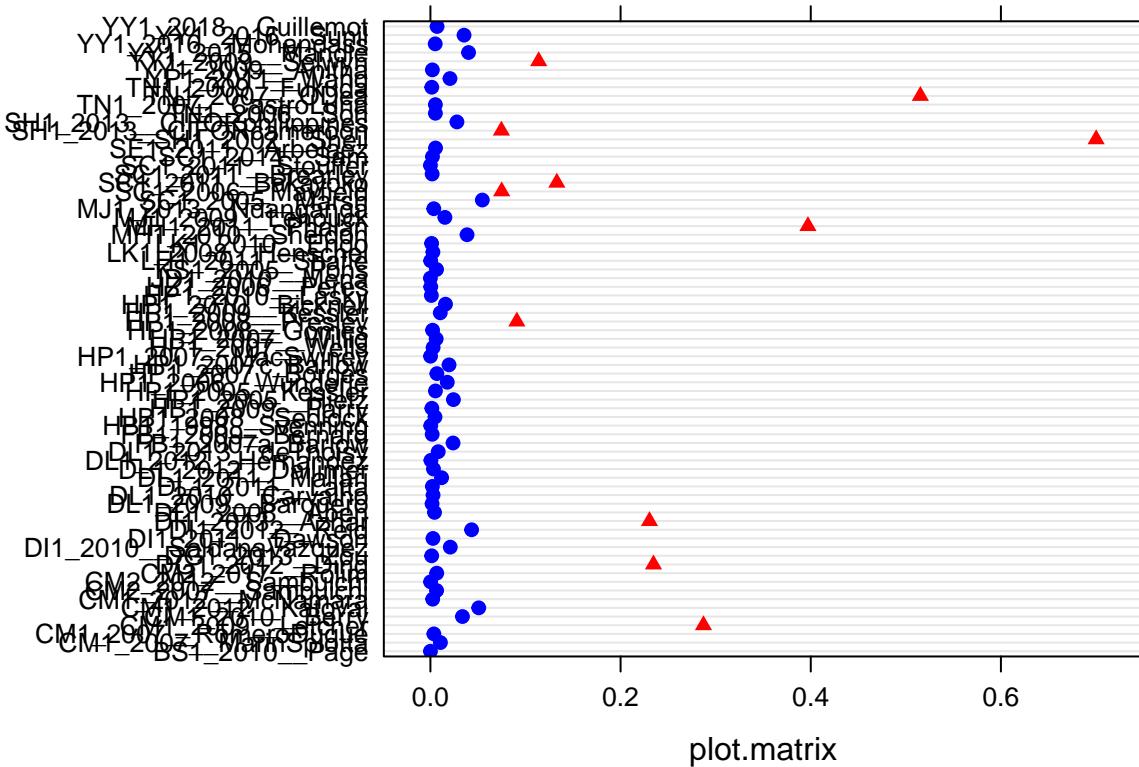
Simpson's diversity

Cut off Cook's distance = $4/73 = 0.055$

```
# Count number of source IDs that are greater than the cut off
length(cook_sources_simpson$Source_ID)
```

```
## [1] 11
```

```
# Plot the results
influence.ME::plot.estex(x=INFM1_simpson, which = "cook", cutoff = 0.06)
```



Cut off DFBETAS = $2/\sqrt{73} = 0.23$

```
# Count number of source IDs that are greater than the cut off
length(dfbetas_sources_simpson$Source_ID)
```

```
## [1] 41
```

Robust analysis

Total abundance model

```
# Run the original model
full_model_abundance <- lmer(logAbundance ~ LandUse.0 + Kingdom + LandUse.0*Kingdom +
                               (1|Source_ID) + (1|SS) + (1|SSB), data= abundance)
```

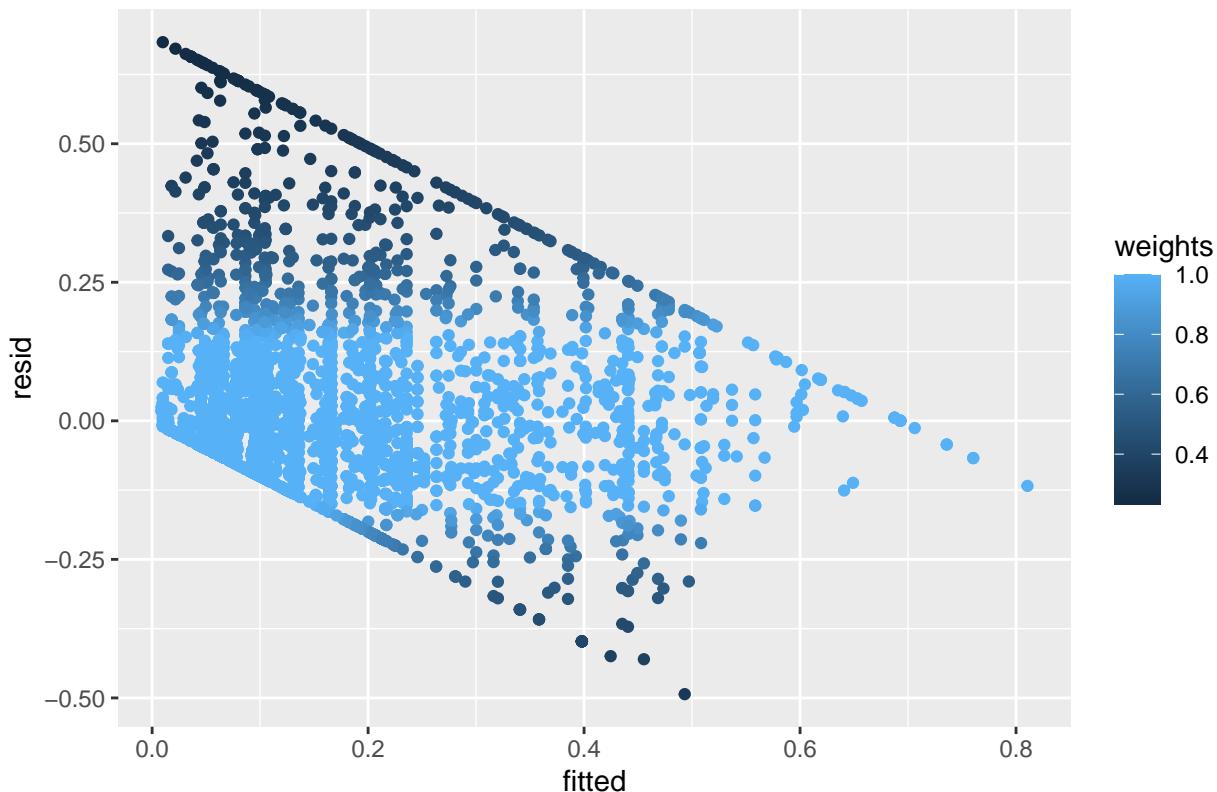
```
# Run the robust analysis
full_model_abundance_robust <- rlmr(logAbundance ~ LandUse.0 + Kingdom + LandUse.0*Kingdom +
                                         (1|Source_ID) + (1|SS) + (1|SSB), data = abundance)
```

Residuals of the robust model

```
plot(full_model_abundance_robust)[1]
```

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

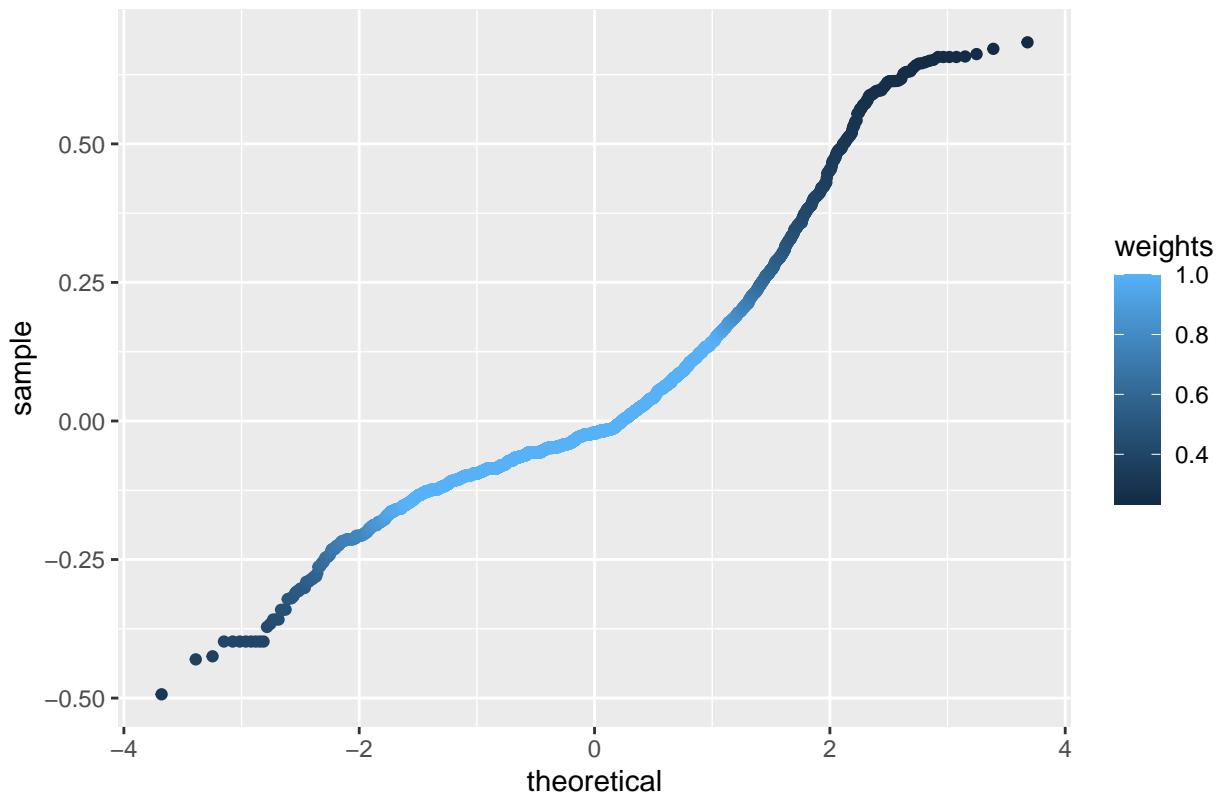
Fitted Values vs. Residuals



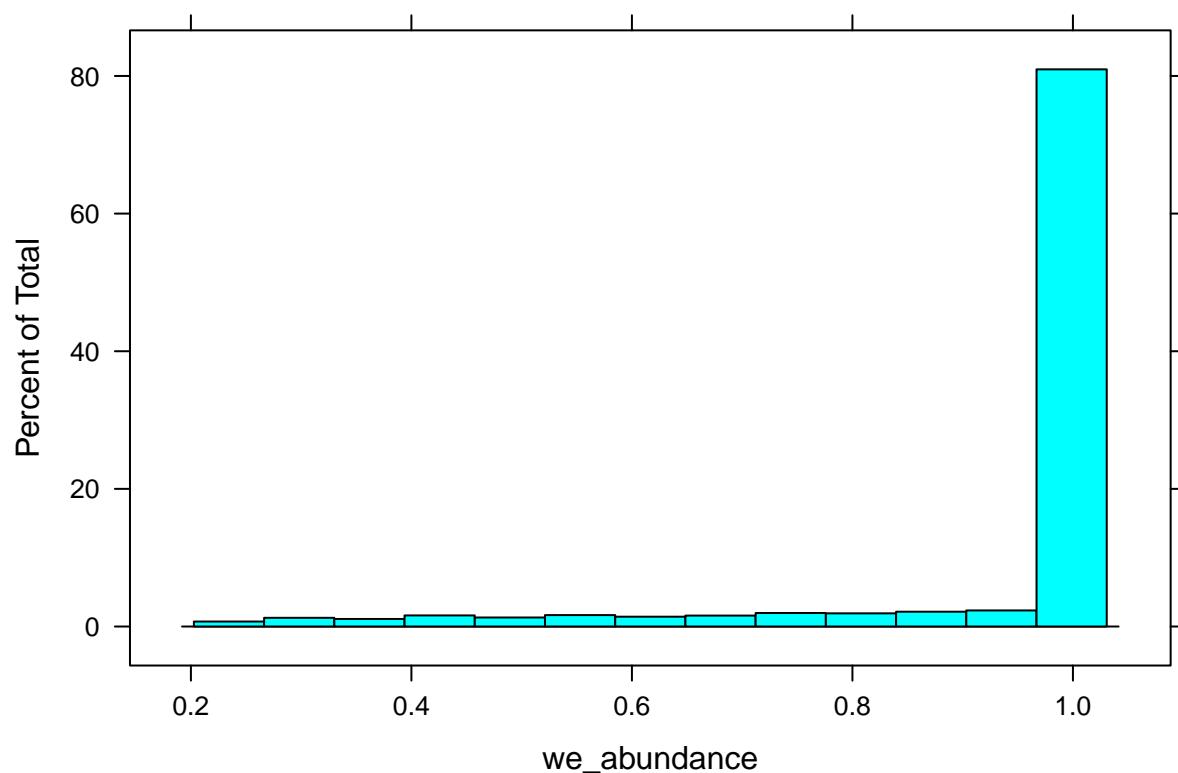
```
plot(full_model_abundance_robust)[2]
```

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

Normal Q–Q vs. Residuals



```
# Get the robustness weights for the sites
we_abundance <- getME(full_model_abundance_robust, "w_e")
# histogram of the robustness weights of sites
histogram(we_abundance)
```

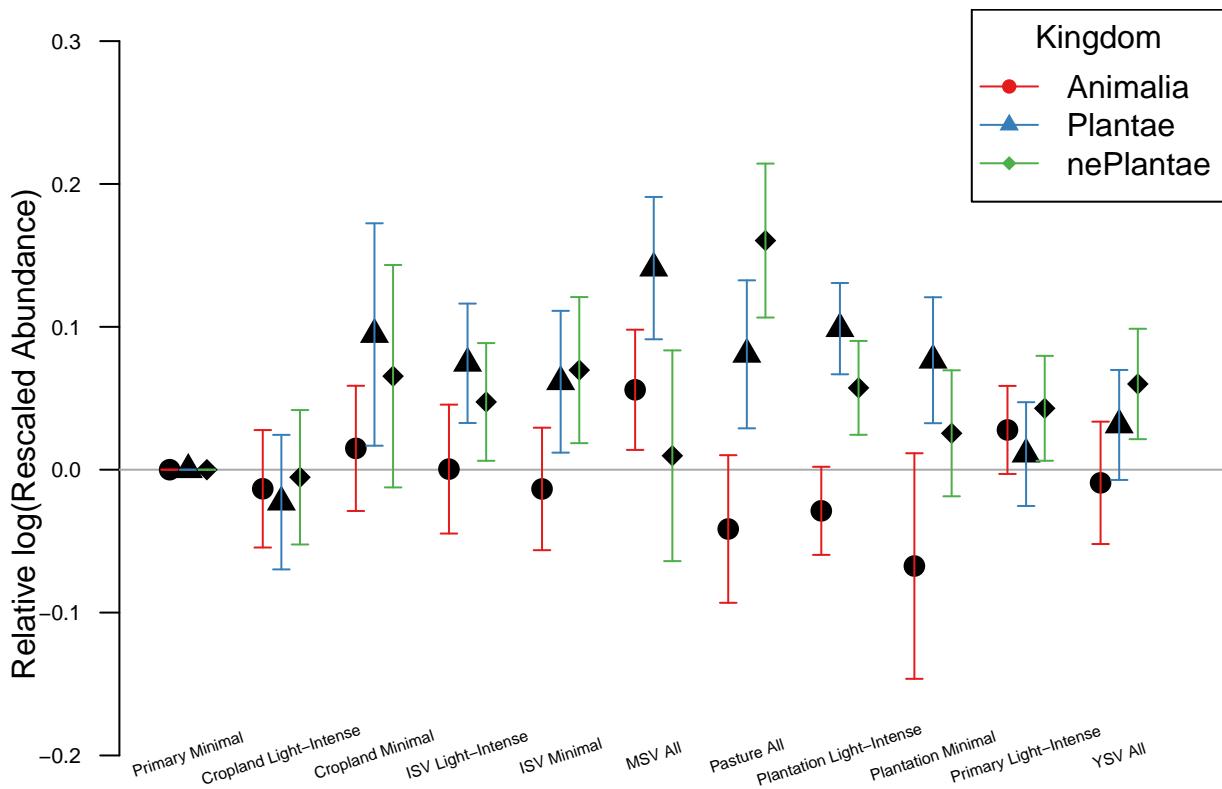


Results of both models:

Original model

```
## Loading required package: roquefort

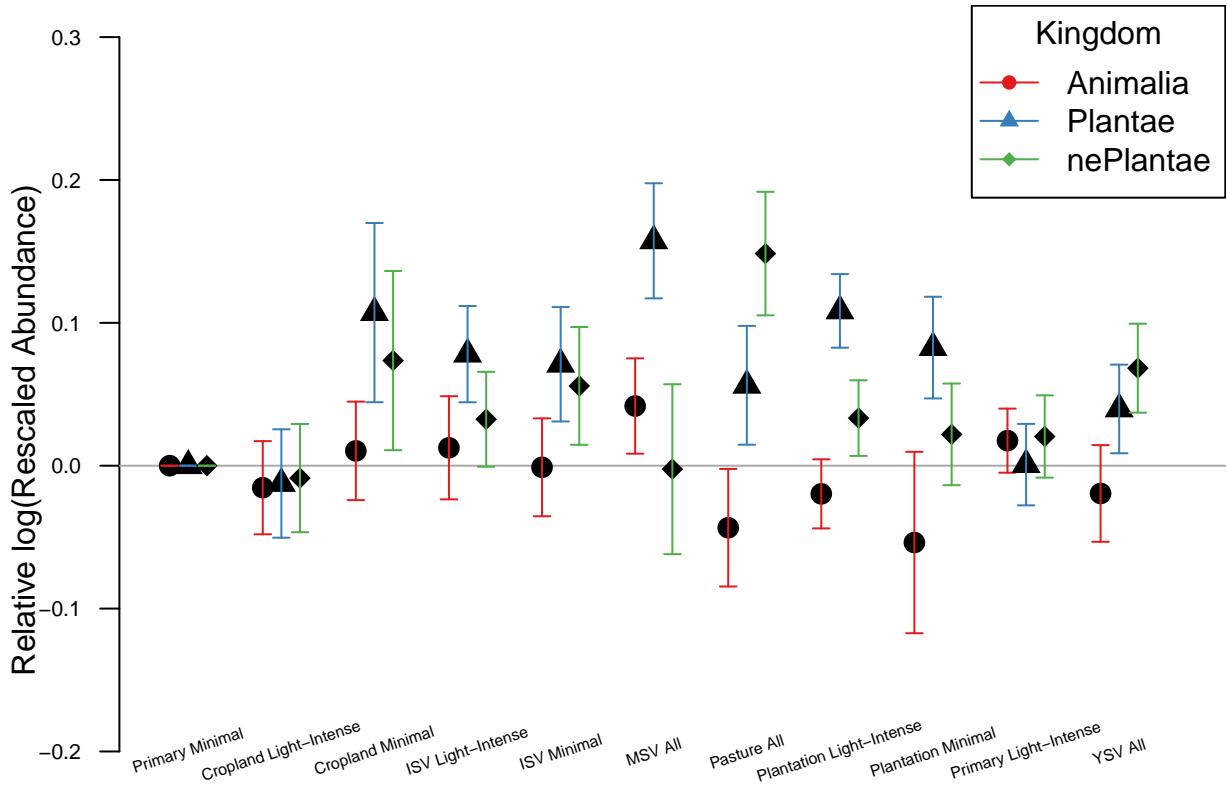
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'roquefort'
```



Robust model

```
## Loading required package: roquefort

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'roquefort'
```



Species Richness model

```
# Run the original model
full_model_richness <- glmer(Species_richness ~ LandUse.0 + Kingdom + LandUse.0:Kingdom +
                               (1|Source_ID) + (1|SS) + (1|SSB) + (1|SSBS),
                               data = sp_richness, family = poisson,
                               control =
                                 glmerControl(optimizer = "bobyqa",
                                               optCtrl = list(maxfun = 20000)))

# Run the robust analysis:
# Since the package only works for gaussian family, I'm going to model sp. richness as log(sp. richness)
# Create the logAbundance column
sp_richness <- mutate(sp_richness, logRichness = log(Species_richness + 1))

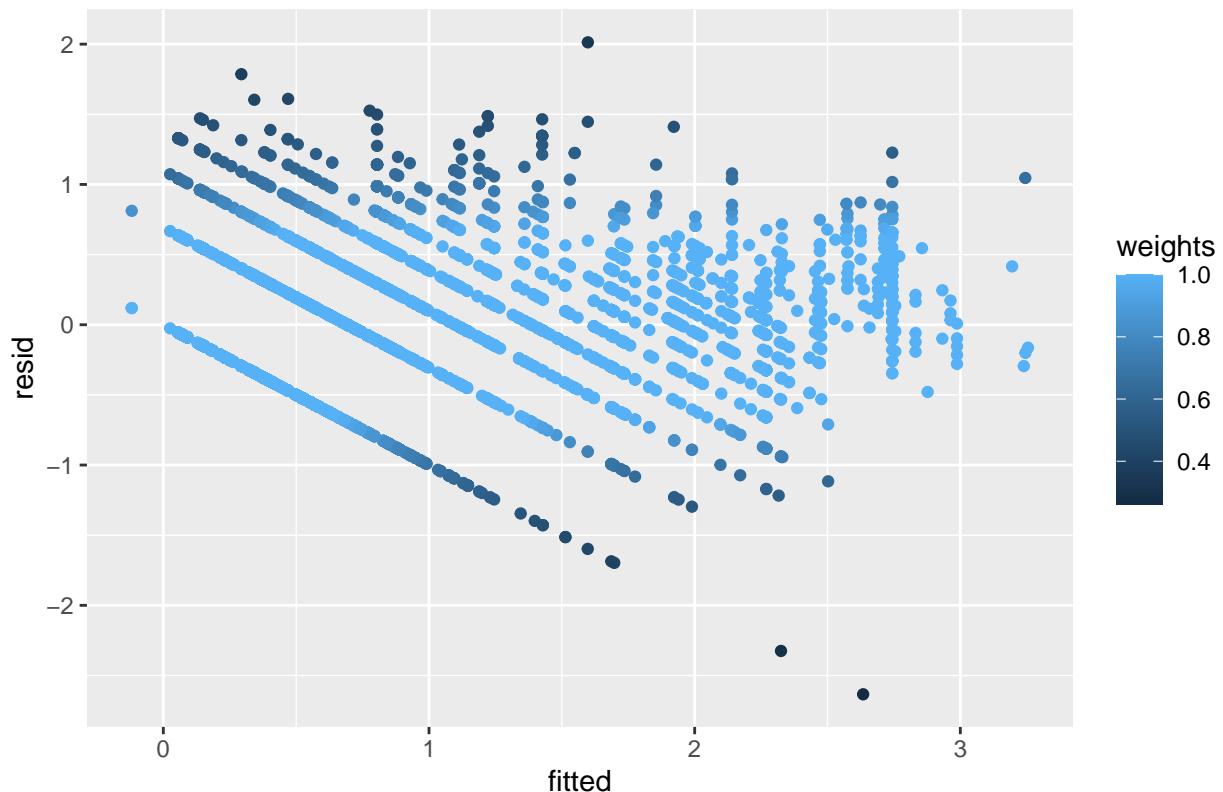
full_model_richness_robust <- rlm(logRichness ~ LandUse.0 + Kingdom + LandUse.0*Kingdom +
                                    (1|Source_ID) + (1|SS) + (1|SSB), data = sp_richness)
```

Residuals of the robust model

```
plot(full_model_richness_robust)[1]
```

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

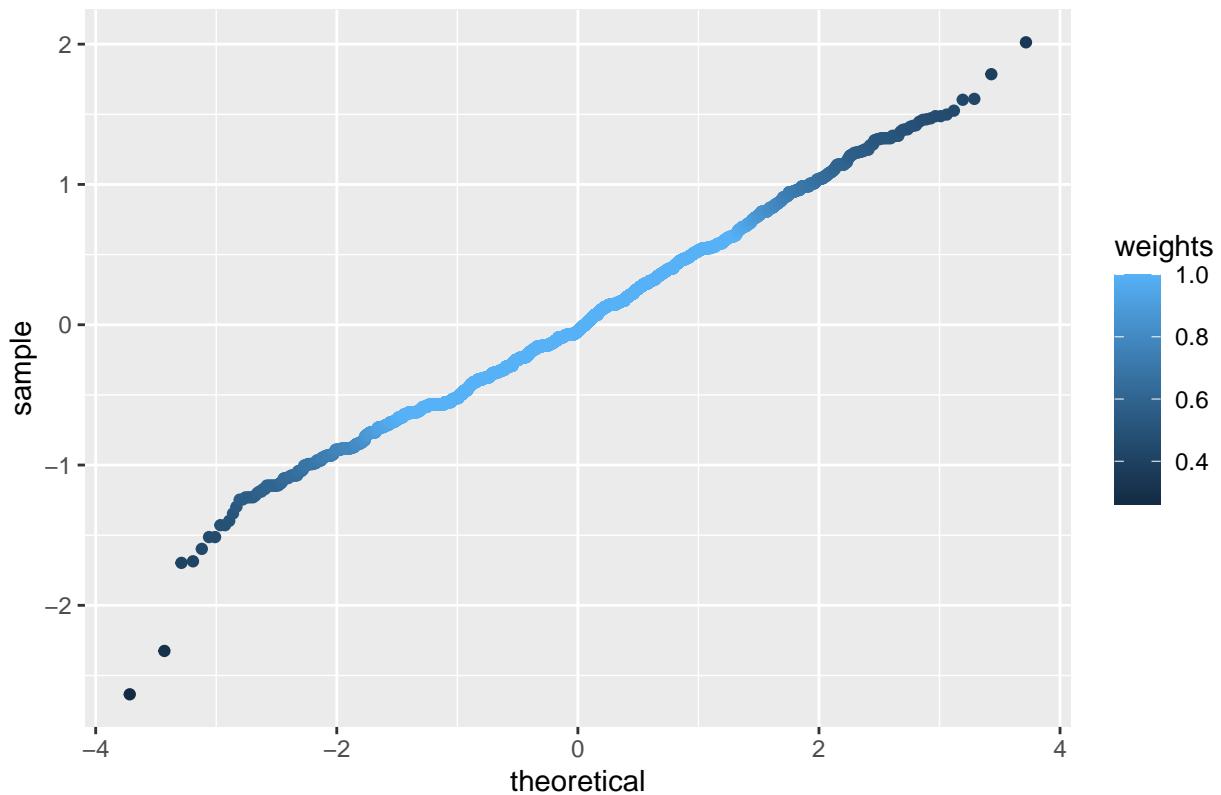
Fitted Values vs. Residuals



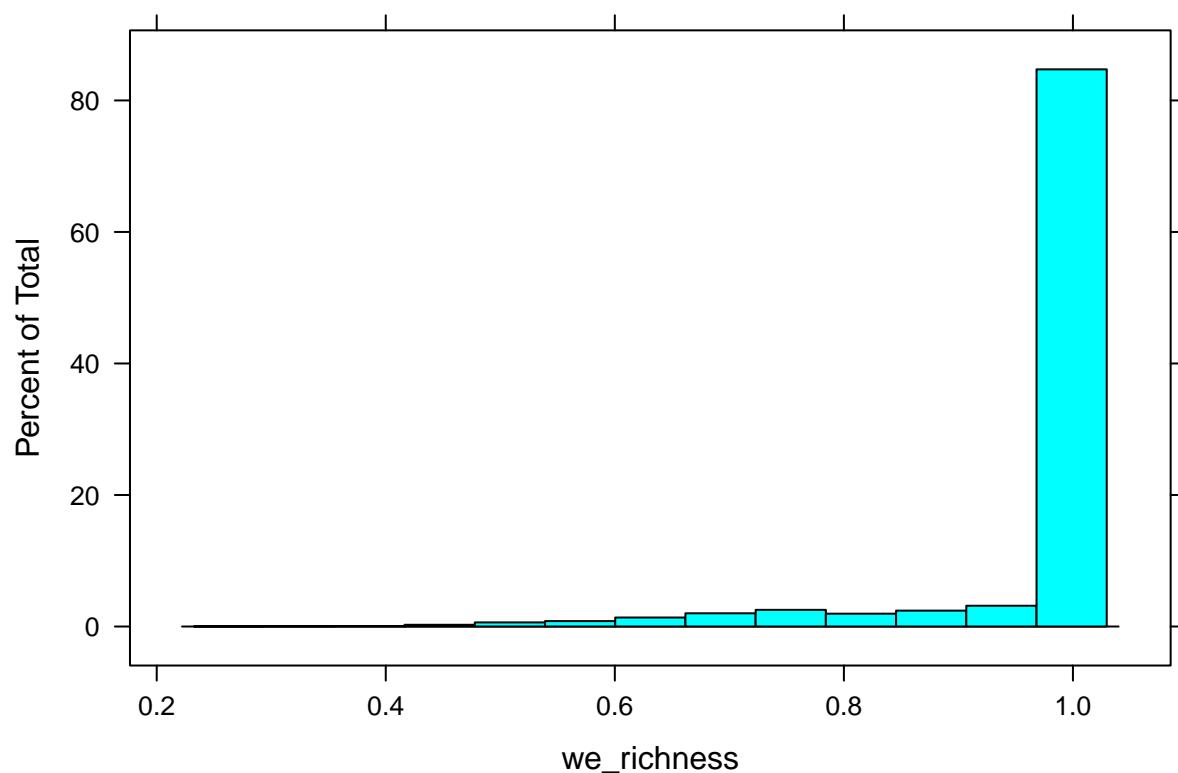
```
plot(full_model_richness_robust)[2]
```

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

Normal Q–Q vs. Residuals



```
# Get the robustness weights for the sites
we_richness <- getME(full_model_richness_robust, "w_e")
# histogram of the robustness weights of sites
histogram(we_richness)
```

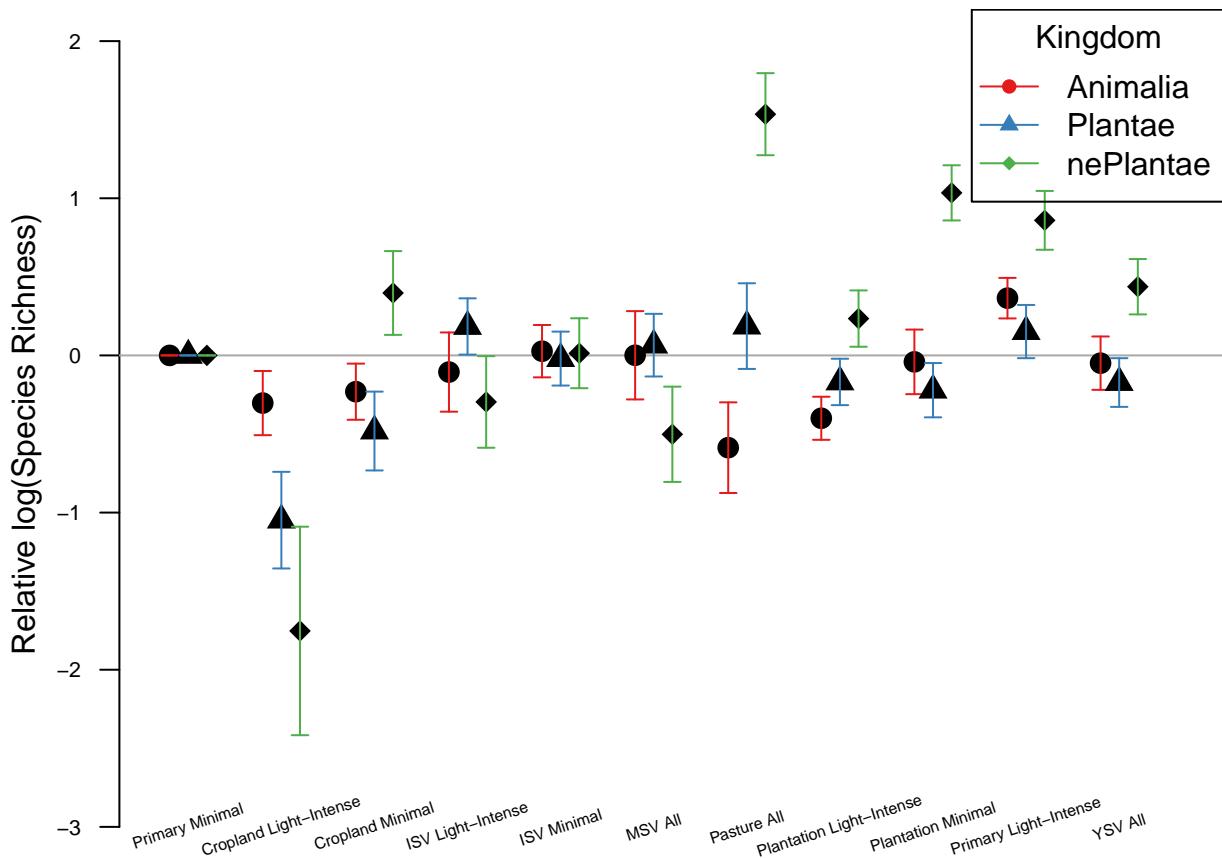


Results of both models:

Original model

```
## Loading required package: roquefort

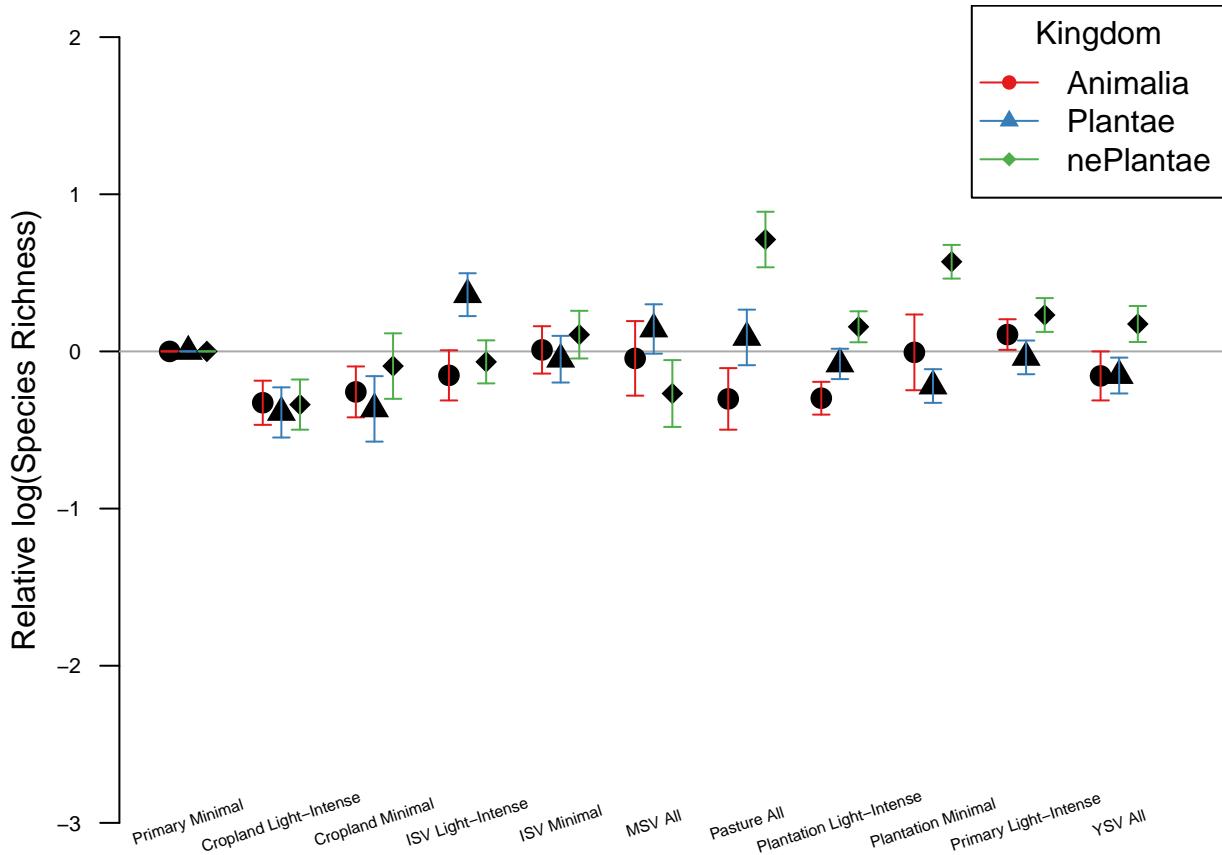
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'roquefort'
```



Robust model

```
## Loading required package: roquefort

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'roquefort'
```



Simpson's diversity model

```
# Run the original model
full_model_simpson <- lmer(log_one_over_D ~ LandUse.0 + Kingdom + LandUse.0:Kingdom +
  (1|Source_ID) + (1|SS) + (1|SSB),
  data = simpson)
```

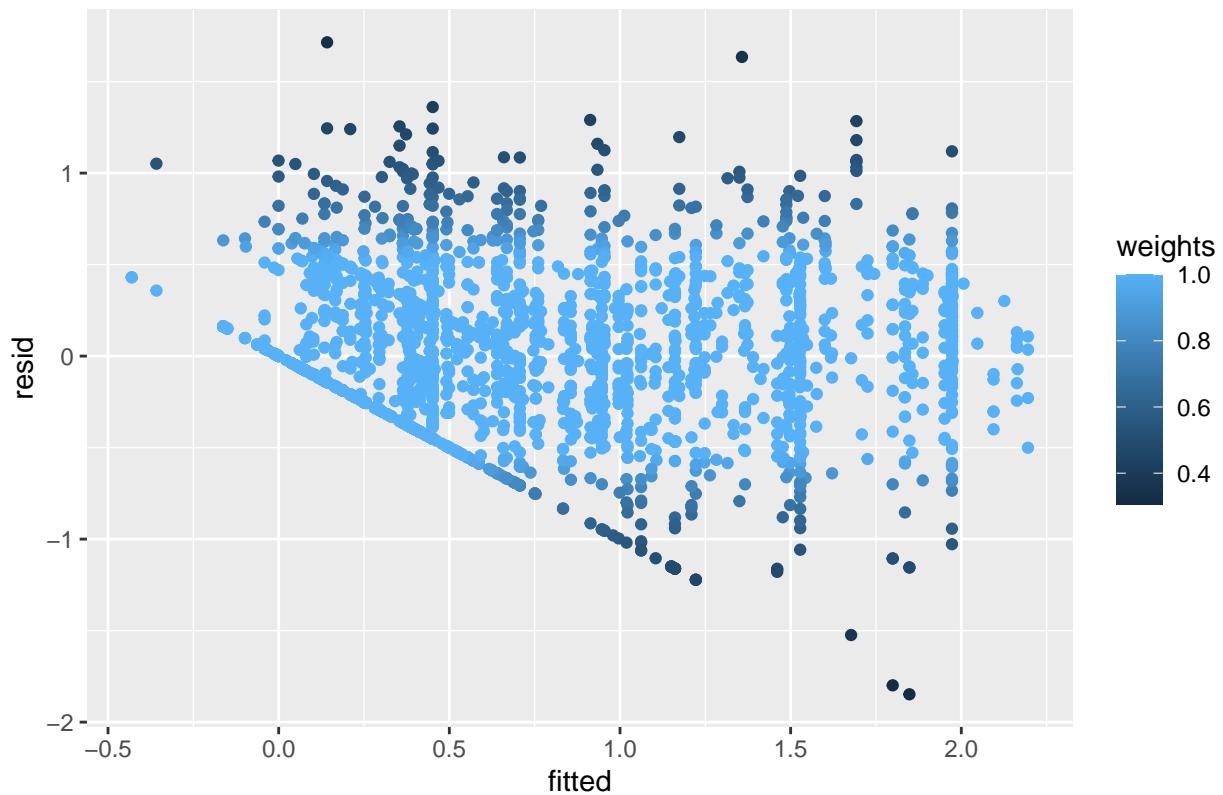
```
# Run the robust analysis:
full_model_simpson_robust <- rlmr(log_one_over_D ~ LandUse.0 + Kingdom + LandUse.0:Kingdom +
  (1|Source_ID) + (1|SS) + (1|SSB),
  data = simpson)
```

Residuals of the robust model

```
plot(full_model_simpson_robust) [1]
```

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

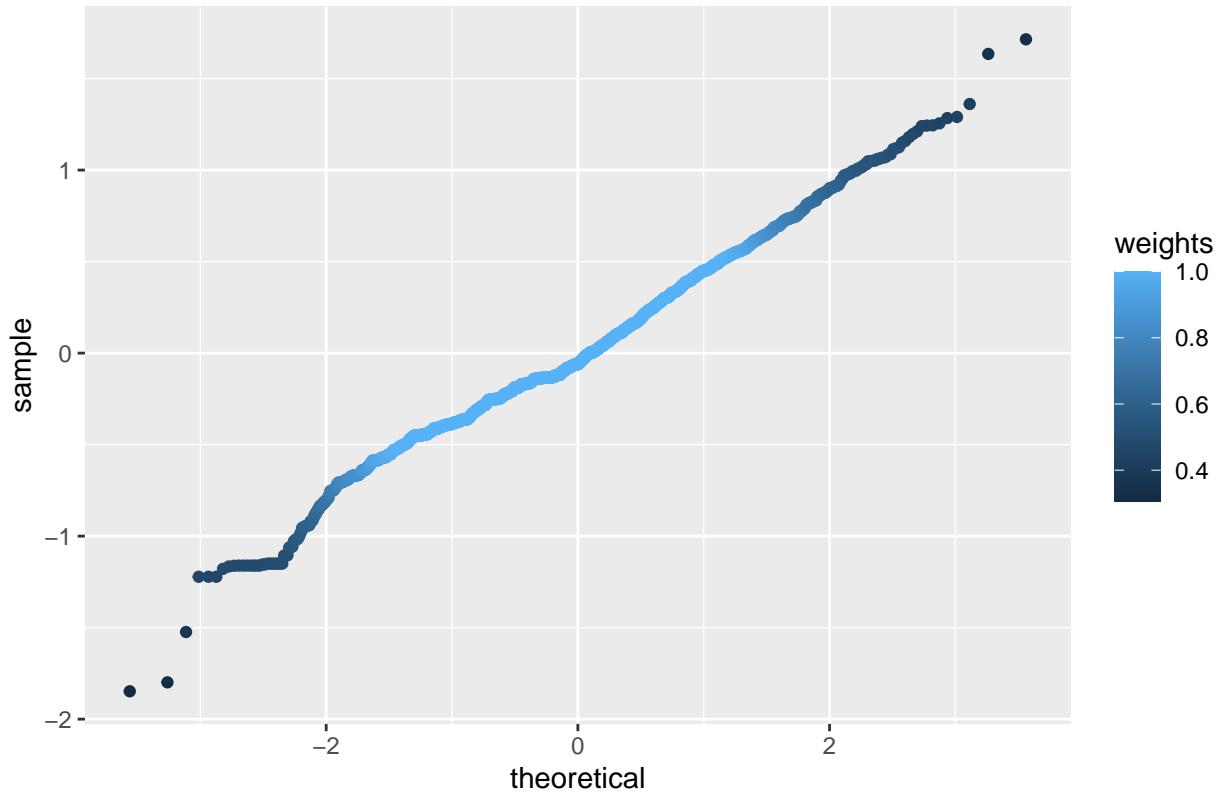
Fitted Values vs. Residuals



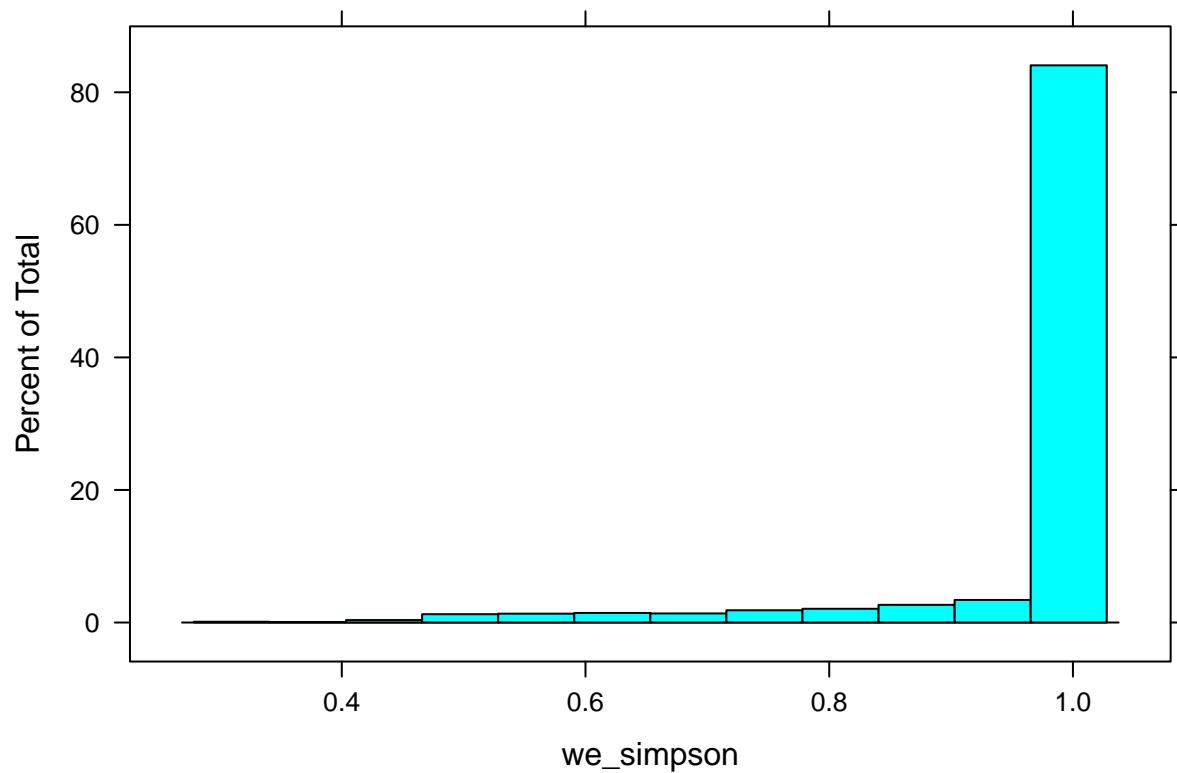
```
plot(full_model_simpson_robust)[2]
```

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

Normal Q–Q vs. Residuals



```
# Get the robustness weights for the sites
we_simpson <- getME(full_model_simpson_robust, "w_e")
# histogram of the robustness weights of sites
histogram(we_simpson)
```

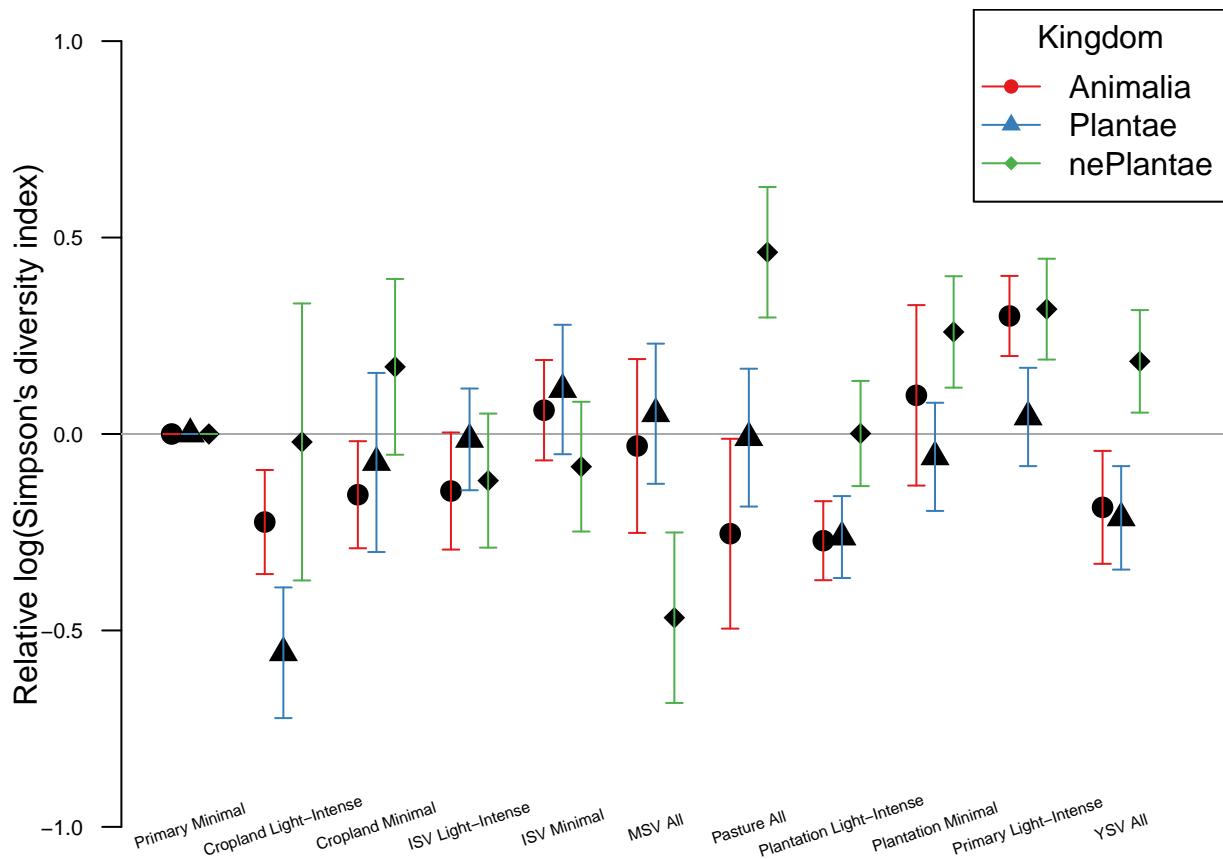


Results of both models:

Original model

```
## Loading required package: roquefort

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'roquefort'
```



Robust model

```
## Loading required package: roquefort

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'roquefort'
```

