```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.1.3
```

```
library(emmeans)
```

```
## Warning: package 'emmeans' was built under R version 4.1.3
```

```
library(lme4)
```

```
## Warning: package 'lme4' was built under R version 4.1.3
```

```
## Loading required package: Matrix
```

```
library(stats)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library(DHARMa)
```

```
## Warning: package 'DHARMa' was built under R version 4.1.3
```

```
## This is DHARMa 0.4.6. For overview type '?DHARMa'. For recent changes, type news(package = 'DHARMa')
```

```
library(magrittr)
```

```
## Warning: package 'magrittr' was built under R version 4.1.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

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

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

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

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
## Warning: package 'tibble' was built under R version 4.1.3
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
## Warning: package 'purrr' was built under R version 4.1.3
```

```
## Warning: package 'stringr' was built under R version 4.1.3
```

```
## Warning: package 'forcats' was built under R version 4.1.3
```

```
## Warning: package 'lubridate' was built under R version 4.1.3
```

```
## -- Attaching core tidyverse packages ----------------------- tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.1     v tidyr     1.3.0
## v readr     2.1.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand()    masks Matrix::expand()
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x tidyr::pack()      masks Matrix::pack()
## x purrr::set_names() masks magrittr::set_names()
## x tidyr::unpack()    masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
DNA_extractie_protocollen <- read_excel("F:/UGent/2022-2023/Masterthesis/DNA extractions protocols.xlsx

DNA_extractie_protocollen %>% #Barplots of extraction success depending on method of preservation
  count(Storage, Extraction_success) %>%
  group_by(Storage) %>%
  ggplot() + aes(Storage, n, fill = Extraction_success, label = n) +
  geom_col() +
  geom_text(position=position_stack(0.5)) +
  xlab('Preservation method') +
  ylab('Amount of extractions') +
  scale_fill_manual('DNA extraction', values=c('steelblue','coral2'))
```
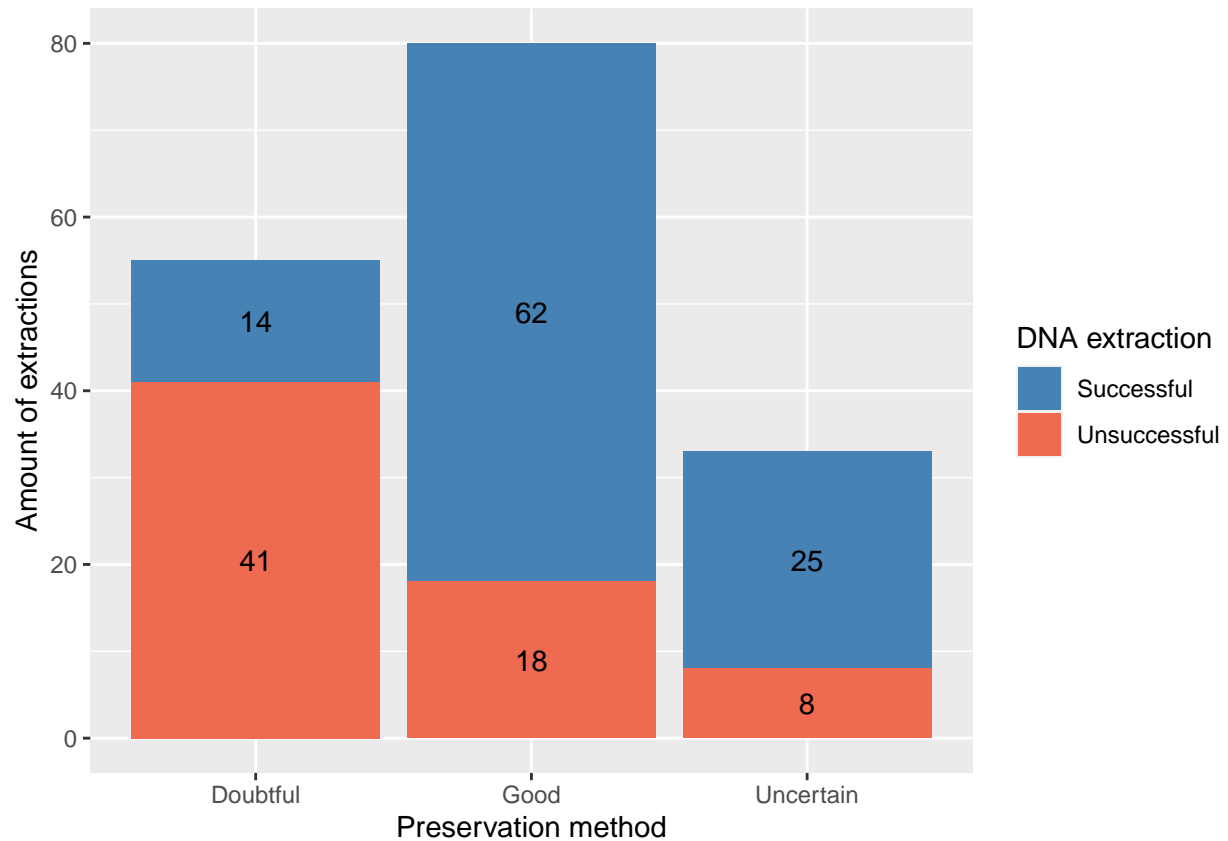
```r
DNA_extractie_protocollen %>% #Barplots of extraction success depending extraction protocol
  count(BSA_and_RepliG, Extraction_success) %>%
  group_by(BSA_and_RepliG) %>%
  ggplot() + aes(BSA_and_RepliG, n, fill = Extraction_success, label = n) +
  geom_col() +
  geom_text(position=position_stack(0.5)) +
  xlab('BSA is added (No/Yes) and which REPLI-g protocol is used') +
  ylab('Amount of extractions') +
  scale_fill_manual('Extraction success', values=c('steelblue','coral2'))
```
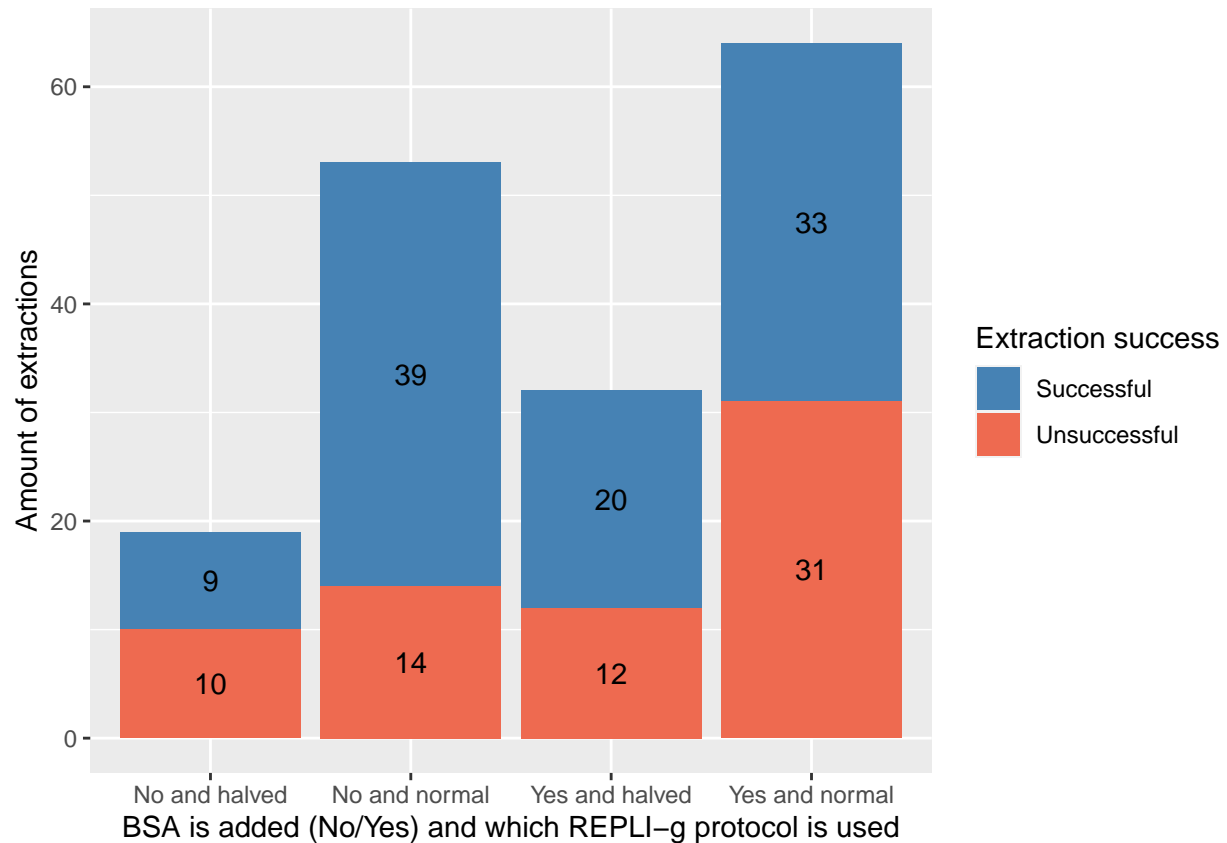
```
#generalized linear model of extraction success  depending on the extraction protocol
full <- glmer(Extraction_success_bin~BSA_added*RepliG+Storage+(1|Species), family = binomial, data=DNA_
summary(full)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Extraction_success_bin ~ BSA_added * RepliG + Storage + (1 |
##      Species)
##    Data: DNA_extractie_protocollen
##
##      AIC      BIC   logLik deviance df.resid
##    184.8    206.7    -85.4    170.8      161
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2737 -0.4104  0.2893  0.4741  1.7300
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  Species (Intercept) 1.742    1.32
## Number of obs: 168, groups:  Species, 50
##
## Fixed effects:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -3.0775     0.9227  -3.335 0.000852 ***
```

4

```
## BSA_addedYes                   2.6003      0.9686   2.685 0.007258 **
## RepliGnormal                   1.7383      0.7543   2.305 0.021188 *
## StorageGood                    3.1487      0.6752   4.663 3.11e-06 ***
## StorageUncertain               3.5380      0.9716   3.641 0.000271 ***
## BSA_addedYes:RepliGnormal      -2.7913     1.0494  -2.660 0.007816 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) BSA_dY RplGnr StrgGd StrgUn
## BSA_addedYs -0.804
## RepliGnorml -0.689  0.640
## StorageGood -0.699  0.540  0.215
## StorgUncrtn -0.611  0.490  0.279  0.638
## BSA_dddY:RG  0.640 -0.838 -0.775 -0.333 -0.410
```

```
emmeans(full, specs = pairwise ~ Storage, type = "respons")
```

```
## $emmeans
##  Storage    prob     SE  df asymp.LCL asymp.UCL
##  Doubtful  0.167 0.0731 Inf    0.0669     0.360
##  Good      0.824 0.0657 Inf    0.6581     0.919
##  Uncertain 0.873 0.0843 Inf    0.6076     0.969
##
## Results are averaged over the levels of: BSA_added, RepliG
## Confidence level used: 0.95
## Intervals are back-transformed from the logit scale
##
## $contrasts
##  contrast            odds.ratio     SE  df null z.ratio p.value
##  Doubtful / Good         0.0429 0.0290 Inf    1  -4.663  <.0001
##  Doubtful / Uncertain    0.0291 0.0282 Inf    1  -3.641  0.0008
##  Good / Uncertain        0.6775 0.5081 Inf    1  -0.519  0.8620
##
## Results are averaged over the levels of: BSA_added, RepliG
## P value adjustment: tukey method for comparing a family of 3 estimates
## Tests are performed on the log odds ratio scale
```

```
emmeans(full, specs = pairwise ~ BSA_added*RepliG, type = "respons")
```

```
## $emmeans
##  BSA_added RepliG prob     SE  df asymp.LCL asymp.UCL
##  No        halved 0.300 0.140 Inf     0.104     0.613
##  Yes       halved 0.852 0.088 Inf     0.595     0.958
##  No        normal 0.709 0.106 Inf     0.471     0.869
##  Yes       normal 0.668 0.103 Inf     0.447     0.833
##
## Results are averaged over the levels of: Storage
## Confidence level used: 0.95
## Intervals are back-transformed from the logit scale
##
## $contrasts
##  contrast              odds.ratio     SE  df null z.ratio p.value
```
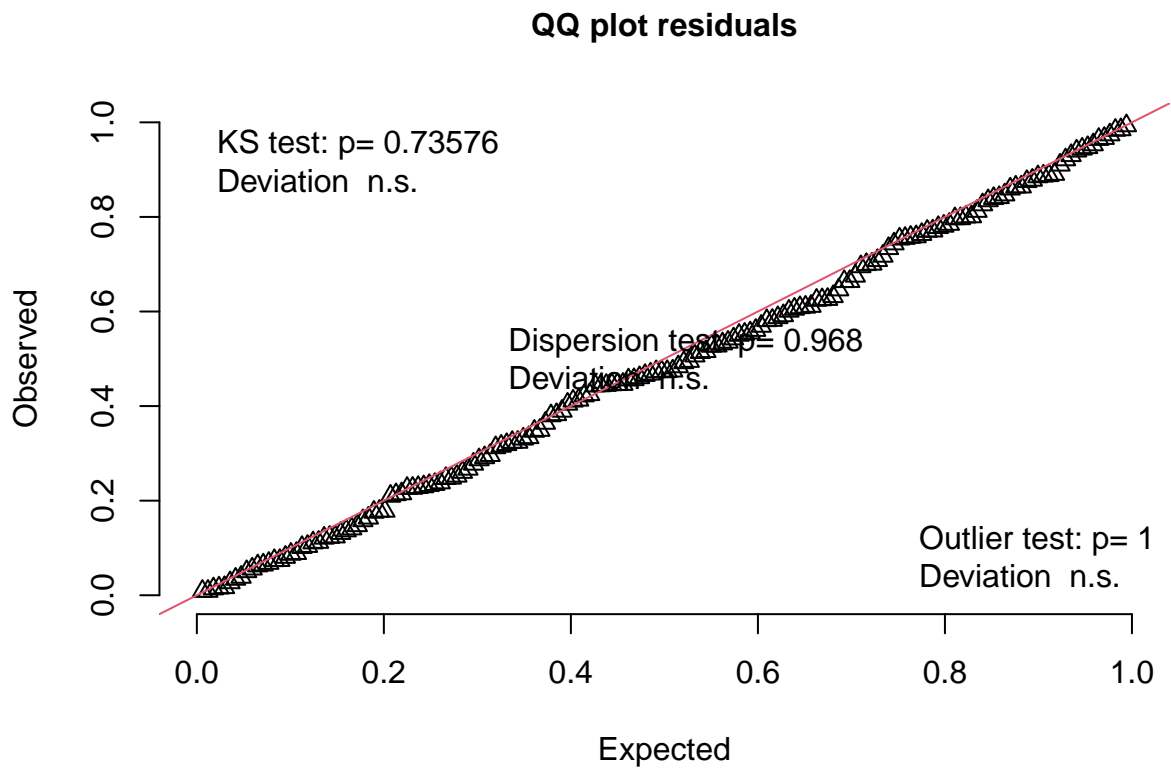
```
##  No halved / Yes halved      0.0742 0.0719 Inf    1  -2.685  0.0365
##  No halved / No normal       0.1758 0.1326 Inf    1  -2.305  0.0969
##  No halved / Yes normal      0.2128 0.1666 Inf    1  -1.977  0.1968
##  Yes halved / No normal      2.3679 1.7910 Inf    1   1.140  0.6649
##  Yes halved / Yes normal     2.8663 1.9086 Inf    1   1.581  0.3893
##  No normal / Yes normal      1.2105 0.7013 Inf    1   0.330  0.9876
##
## Results are averaged over the levels of: Storage
## P value adjustment: tukey method for comparing a family of 4 estimates
## Tests are performed on the log odds ratio scale
```

```
#test assumptions for glmer
testDispersion(full)
```
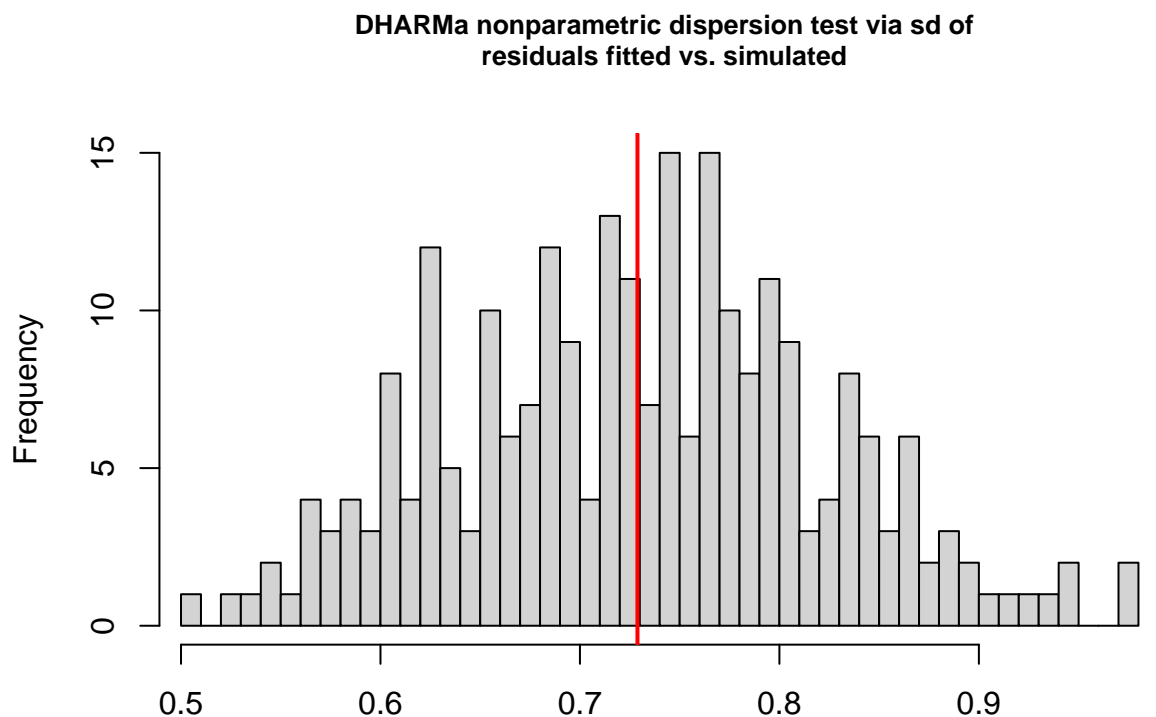
**DHARMa nonparametric dispersion test via sd of
residuals fitted vs. simulated**

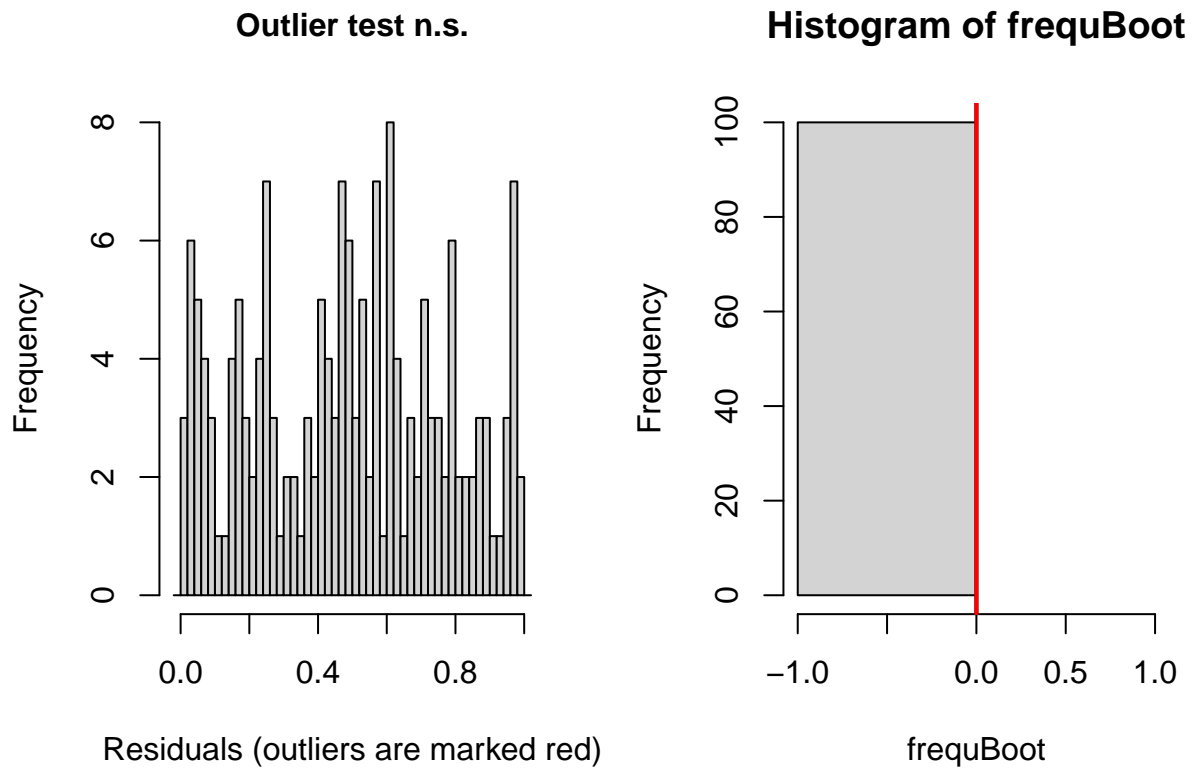Simulated values, red line = fitted model. p–value (two.sided) = 0.968

```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.99935, p-value = 0.968
## alternative hypothesis: two.sided
```

```
testUniformity(full)
```

QQ plot residuals

KS test: p= 0.73576
Deviation n.s.

Dispersion test: p= 0.968
Deviation n.s.

Outlier test: p= 1
Deviation n.s.

Observed

Expected

```
##
##   One-sample Kolmogorov-Smirnov test
##
## data:  simulationOutput$scaledResiduals
## D = 0.052862, p-value = 0.7358
## alternative hypothesis: two-sided
```

```
testOutliers(full)
```

Outlier test n.s. **Histogram of frequBoot**

Frequency (left plot y-axis: 0, 2, 4, 6, 8; x-axis: 0.0, 0.4, 0.8)

Residuals (outliers are marked red)

Frequency (right plot y-axis: 0, 20, 40, 60, 80, 100; x-axis: −1.0, 0.0, 0.5, 1.0)

frequBoot

```
## 
##   DHARMa bootstrapped outlier test
## 
## data:  full
## outliers at both margin(s) = 0, observations = 168, p-value = 1
## alternative hypothesis: two.sided
##  percent confidence interval:
##  0 0
## sample estimates:
## outlier frequency (expected: 0 )
##                                 0
```

```
simulationOutput <- simulateResiduals(fittedModel = full, plot = F)
residuals(simulationOutput, quantileFunction = qnorm, outlierValues = c(-7,7))
```
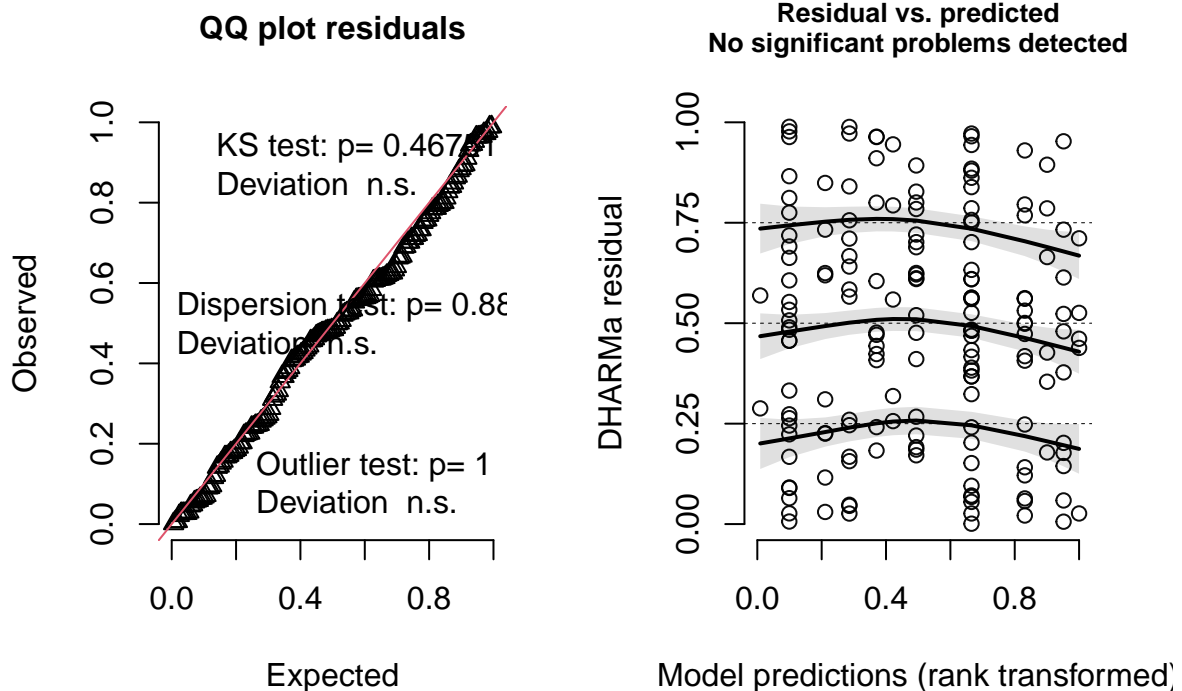
```
##   [1] -0.079292231  1.585778168 -0.703442924  0.696958721 -0.832258911
##   [6] -1.467321537  0.425023922 -0.232691378 -1.935906179  1.801223173
##  [11] -1.596387202 -0.702051910 -0.058563047 -0.458152559  1.173183523
##  [16] -0.904070489  0.151483212 -1.308470296 -1.493509965  1.790737248
##  [21] -3.092608948  0.841396530 -0.168326916  0.677067978  1.344178056
##  [26] -0.921830330  0.148233988  1.815383705  0.792178888 -0.471477659
##  [31] -0.072607966 -0.040107749 -0.184396305  1.250372796 -0.373784503
##  [36] -0.280912797  0.786529942  0.318661283  0.160481744 -0.299493783
##  [41] -0.003506507 -0.067132307 -0.225876418  0.843583134  0.793140555
##  [46]  0.989020493 -1.576662452 -0.654810475  0.277175321 -0.771018706
```

```
## [51] -0.142448220   0.057980618 -1.936317965 -0.338385439   0.493365585
## [56] -0.046835895   0.278973672   1.032274337 -0.041161944   0.621481271
## [61]  0.755867424   1.088974924   0.584795151 -0.046511976   0.309036417
## [66] -1.028747486   0.064814522 -0.338733445   0.159219847 -0.559418828
## [71]  0.173744070   0.498456495 -0.494467080   1.911293555 -0.210827879
## [76]  0.265884099   0.529530550 -0.949475073   0.339151225 -0.059977175
## [81]  0.817736001   0.064394892   1.598766863 -0.193486539   0.318131405
## [86] -1.196039413   0.529544854   0.279002126   1.198320925   0.166450535
## [91]  0.303457058 -0.754880870 -0.750557821 -1.877814918 -1.341121789
## [96] -0.235948552 -0.681125625   0.078350265   0.152858500   0.157025307
## [101] -2.026488237 -0.963630667   0.419329727 -0.207095014   1.475226813
## [106]  0.825823907 -0.154740686   0.017613491 -1.518266533 -0.603031780
## [111] -0.894680278 -1.564416755   0.284398122 -1.008438657   0.944043669
## [116] -0.311980033   0.049851672 -0.925925577   1.240513397   0.557125257
## [121]  0.733879129   0.211995022 -0.110266878   0.693963503   2.003538420
## [126]  1.912175860 -1.931239991   0.884433293 -0.632414892 -0.964124544
## [131]  0.578274530 -0.025703025 -0.689377711   0.554941187 -1.691771856
## [136] -0.686500005 -1.664515081   0.621403868 -2.510480266 -0.833791307
## [141] -1.060552889 -2.488261129   2.285474540   1.791936461   1.107164816
## [146] -0.760592618   2.272775704   0.431707589 -0.644577151   0.361111761
## [151]  0.132822549 -1.334213317 -0.433622492   0.270294277 -0.097257423
## [156]  0.001503791 -1.073981452 -0.872240737   1.671665425   0.289080821
## [161] -0.050934684   0.996541481 -1.523103690 -1.169034412 -1.938719283
## [166]  0.083629127 -0.108463114 -0.621450629
```

```
plot(simulationOutput)
```

DHARMa residual

QQ plot residuals — KS test: p= 0.467; Deviation n.s.; Dispersion test: p= 0.88; Deviation n.s.; Outlier test: p= 1; Deviation n.s. (Observed vs Expected)

Residual vs. predicted / No significant problems detected (DHARMa residual vs Model predictions (rank transformed))
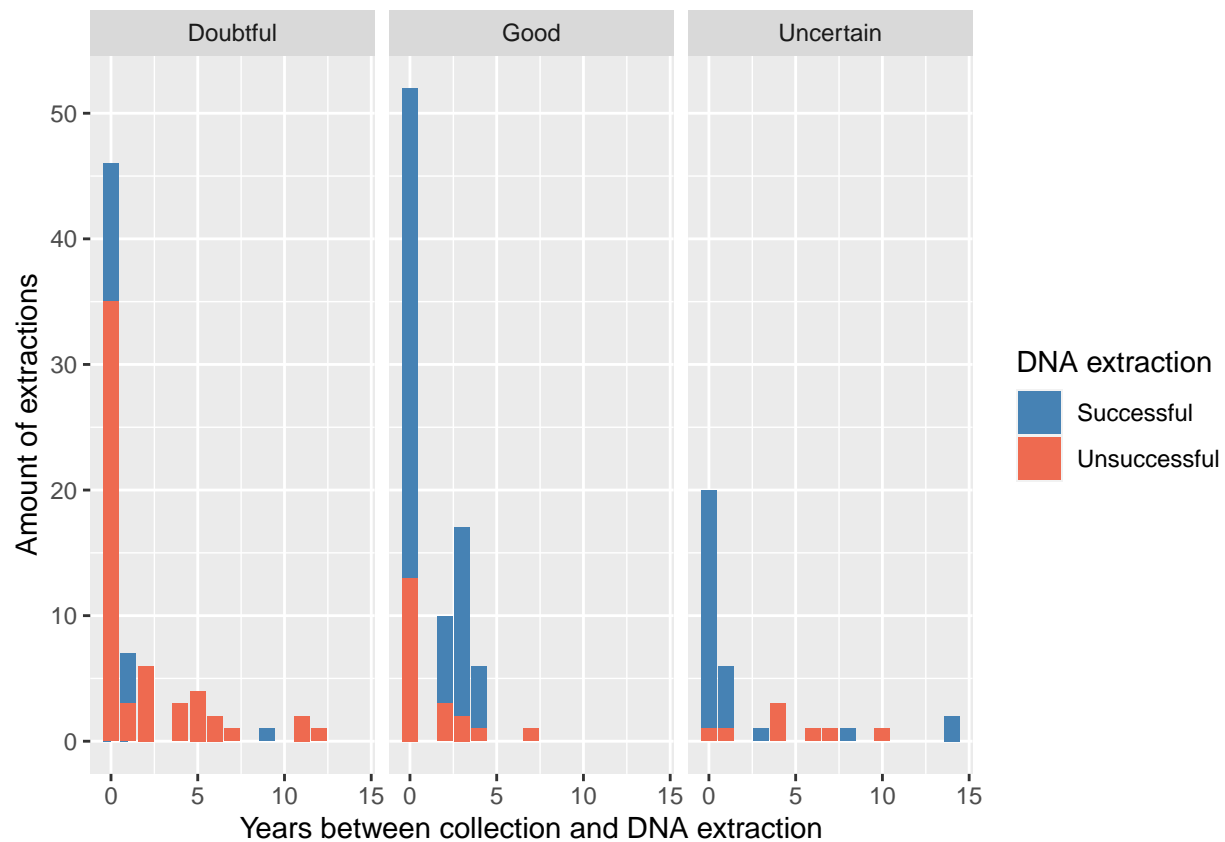
```
extractions_over_time <- read_excel("F:/UGent/2022-2023/Masterthesis/DNA extractions over time.xlsx")

extractions_over_time %>% #Barplots of extractions success depending on the years between collection an
  count(Extraction_success, Storage, Extraction_made_years_after_collection) %>%
  group_by(Extraction_made_years_after_collection) %>%
  ggplot() + aes(Extraction_made_years_after_collection, n, fill = Extraction_success) +
  geom_bar(position="dodge", stat="identity") +
  geom_col() +
  facet_grid(.~Storage) +
  xlab('Years between collection and DNA extraction') +
  ylab('Amount of extractions') +
  scale_fill_manual('DNA extraction', values=c('steelblue','coral2'))
```
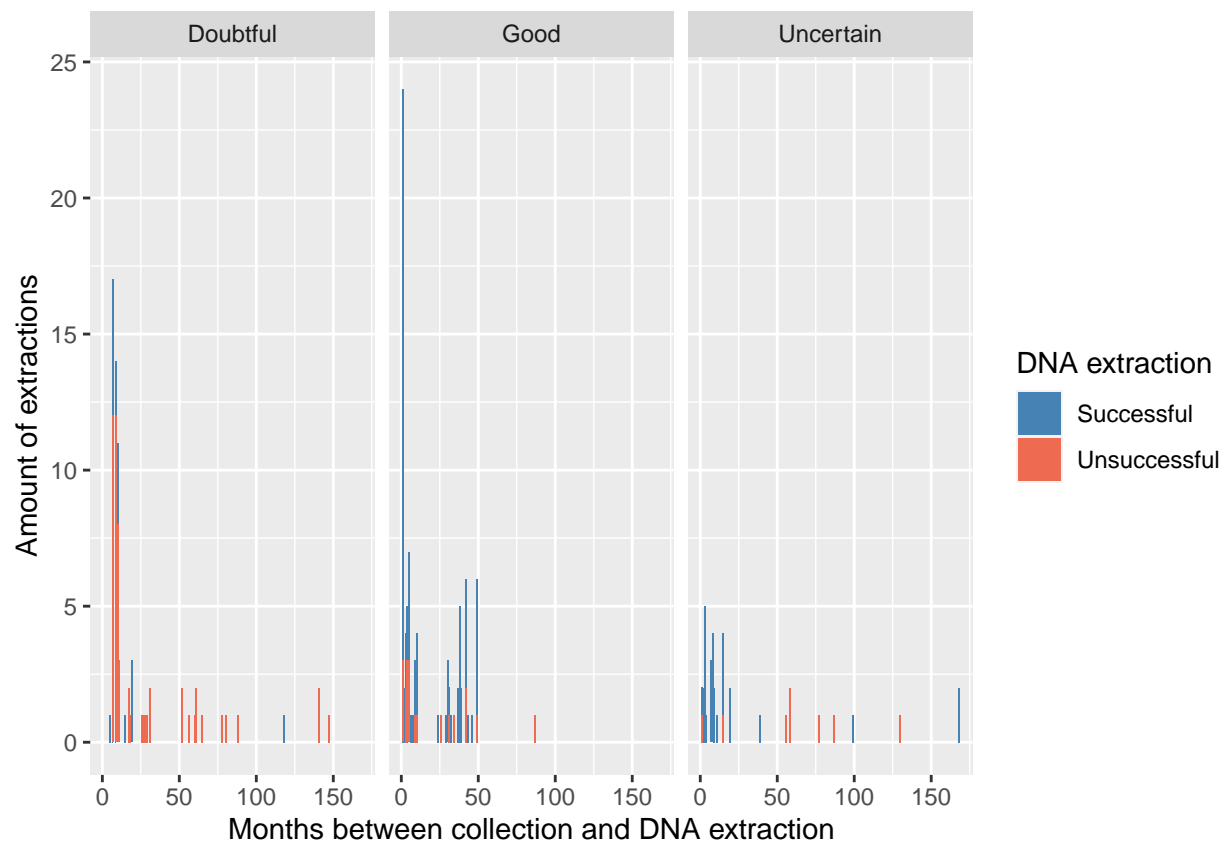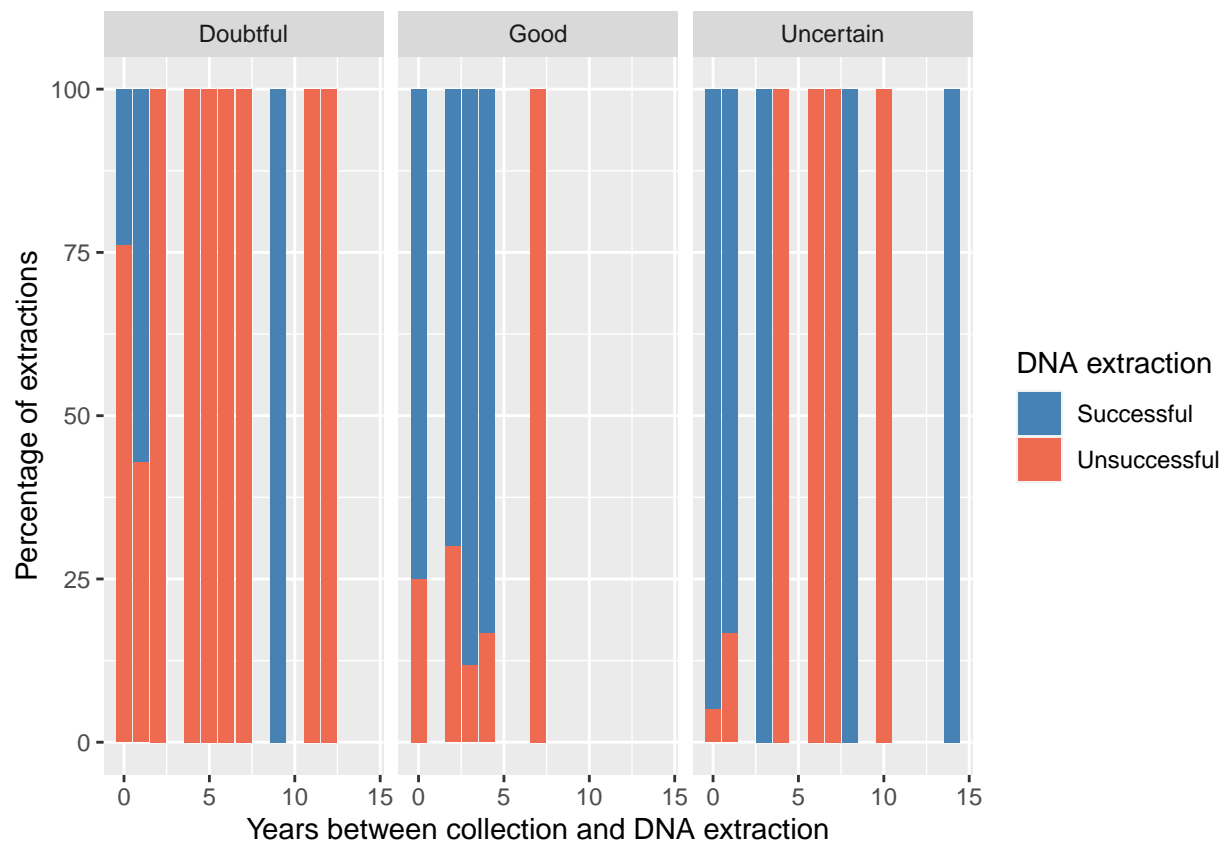


```
extractions_over_time %>% #Barplots of extractions success depending on the months between collection a
  count(Extraction_success, Storage, Extraction_made_months_after_collection) %>%
  group_by(Extraction_made_months_after_collection) %>%
  ggplot() + aes(Extraction_made_months_after_collection, n, fill = Extraction_success) +
  geom_bar(position="dodge", stat="identity") +
  geom_col() +
  facet_grid(.~Storage) +
  xlab('Months between collection and DNA extraction') +
  ylab('Amount of extractions') +
  scale_fill_manual('DNA extraction', values=c('steelblue','coral2'))
```
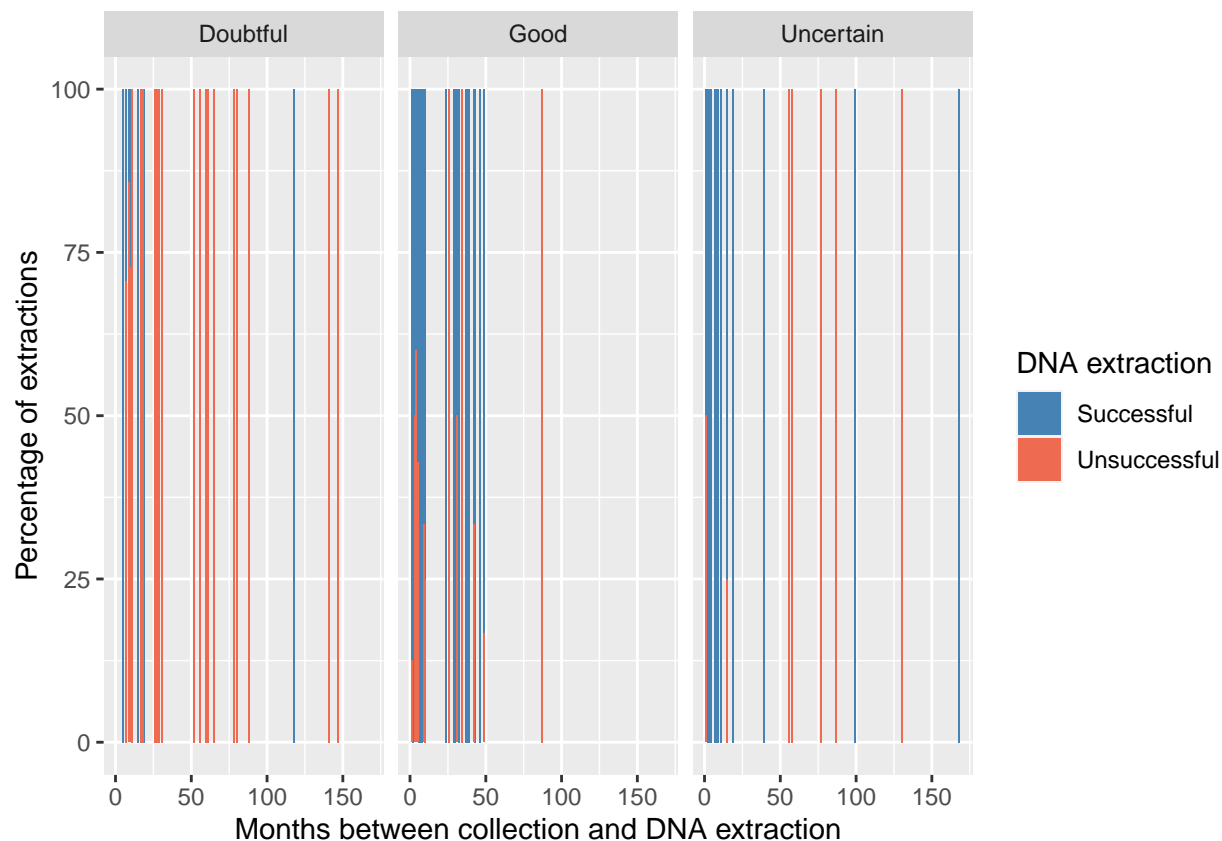
```
extractions_over_time %>% #Barplots of percentage of extraction success depending on the years between
  count(Extraction_success, Storage, Extraction_made_years_after_collection) %>%
  group_by(Extraction_made_years_after_collection, Storage) %>%
  mutate(pct= prop.table(n) * 100) %>%
  ggplot() + aes(Extraction_made_years_after_collection, pct, fill=Extraction_success) +
  geom_bar(stat="identity") +
  geom_col() +
  facet_grid(.~Storage) +
  xlab('Years between collection and DNA extraction') +
  ylab('Percentage of extractions') +
  scale_fill_manual('DNA extraction', values=c('steelblue','coral2'))
```

```
extractions_over_time %>% #Barplots of percentage of extraction success depending on the months between
  count(Extraction_success, Storage, Extraction_made_months_after_collection) %>%
  group_by(Extraction_made_months_after_collection, Storage) %>%
  mutate(pct= prop.table(n) * 100) %>%
  ggplot() + aes(Extraction_made_months_after_collection, pct, fill=Extraction_success) +
  geom_bar(stat="identity") +
  geom_col() +
  facet_grid(.~Storage) +
  xlab('Months between collection and DNA extraction') +
  ylab('Percentage of extractions') +
  scale_fill_manual('DNA extraction', values=c('steelblue','coral2'))
```
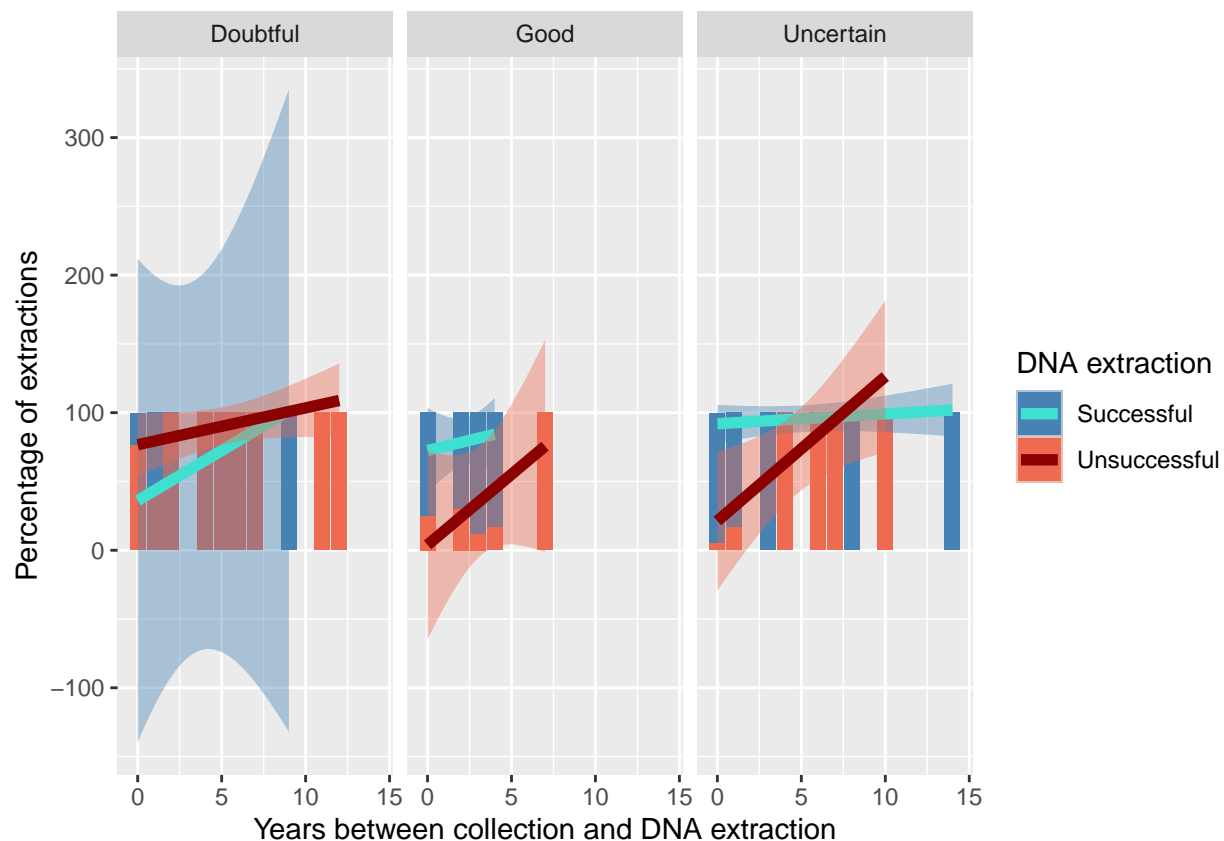
```
extractions_over_time %>% #Barplots of percentage of extraction success depending on the years between
  count(Extraction_success, Storage, Extraction_made_years_after_collection) %>%
  group_by(Extraction_made_years_after_collection, Storage) %>%
  mutate(pct= prop.table(n) * 100) %>%
  ggplot() + aes(Extraction_made_years_after_collection, pct, fill=Extraction_success) +
  geom_bar(stat="identity") +
  geom_col() +
  geom_smooth(aes(x = Extraction_made_years_after_collection, y = pct, color = Extraction_success),
              method = "lm", se = TRUE, size = 2) +
  facet_grid(. ~ Storage) +
  xlab('Years between collection and DNA extraction') +
  ylab('Percentage of extractions') +
  scale_fill_manual('DNA extraction', values = c('steelblue', 'coral2')) +
  scale_color_manual('DNA extraction', values = c('turquoise', 'red4'))
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
## 'geom_smooth()' using formula = 'y ~ x'
```
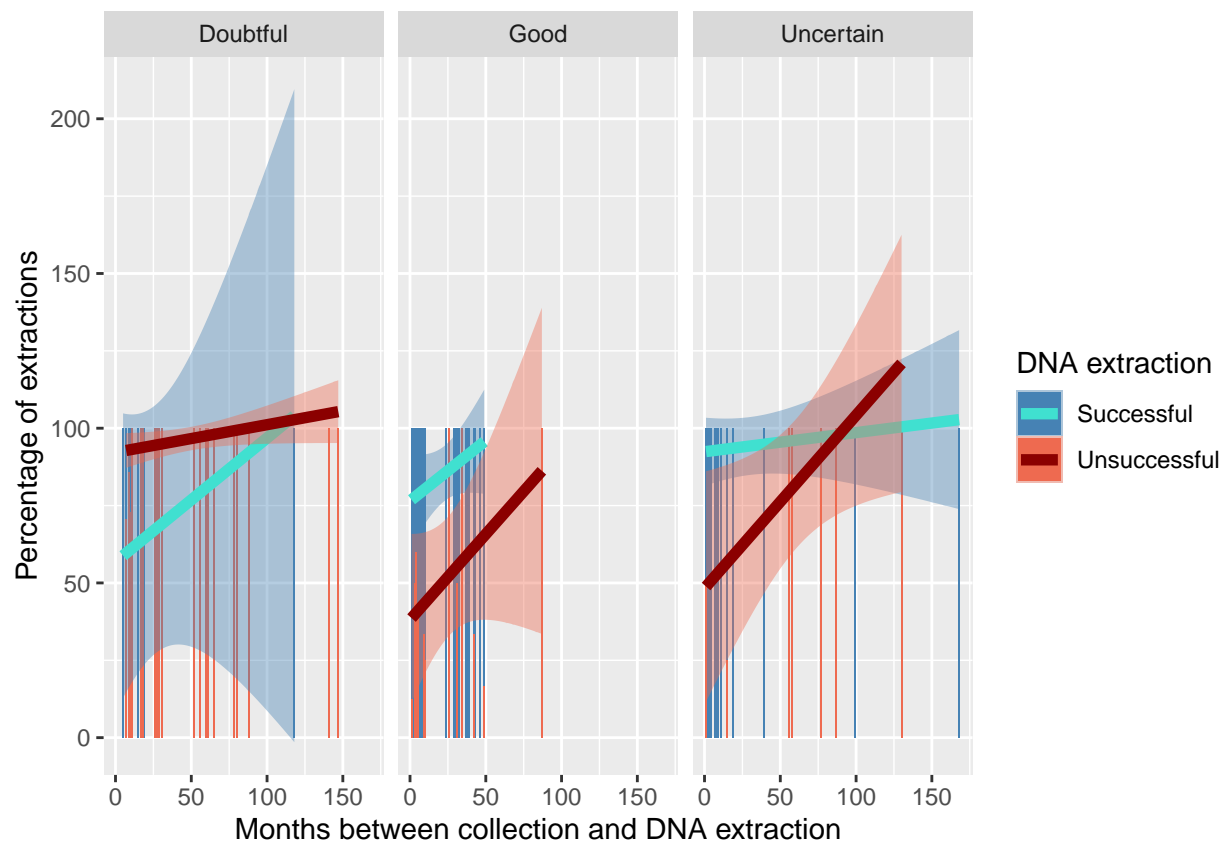
```
extractions_over_time %>% #Barplots of percentage of extraction success depending on the months between
  count(Extraction_success, Storage, Extraction_made_months_after_collection) %>%
  group_by(Extraction_made_months_after_collection, Storage) %>%
  mutate(pct= prop.table(n) * 100) %>%
  ggplot() + aes(Extraction_made_months_after_collection, pct, fill=Extraction_success) +
  geom_bar(stat="identity") +
  geom_col() +
  geom_smooth(aes(x = Extraction_made_months_after_collection, y = pct, color = Extraction_success),
              method = "lm", se = TRUE, size = 2) +
  facet_grid(. ~ Storage) +
  xlab('Months between collection and DNA extraction') +
  ylab('Percentage of extractions') +
  scale_fill_manual('DNA extraction', values = c('steelblue', 'coral2')) +
  scale_color_manual('DNA extraction', values = c('turquoise', 'red4'))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
#generalized linear model of extraction success depending on the time (in months and years) and storage
full <- glmer(Extraction_success_bin~Extraction_made_years_after_collection*Storage+Extraction_made_mon
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0749041 (tol = 0.002, component 1)
```

```
summary(full)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Extraction_success_bin ~ Extraction_made_years_after_collection *
##     Storage + Extraction_made_months_after_collection * Storage +
##     (1 | Species)
##    Data: extractions_over_time
##
##      AIC      BIC   logLik deviance df.resid
##    217.5    250.2    -98.7    197.5      185
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.2256 -0.4973  0.3328  0.5115  2.5070
##
## Random effects:
```

```
##  Groups  Name         Variance Std.Dev.
##  Species (Intercept) 0.6725   0.82
## Number of obs: 195, groups:  Species, 54
##
## Fixed effects:
##                                                         Estimate Std. Error
## (Intercept)                                              -1.3001     1.2147
## Extraction_made_years_after_collection                   -0.3450     1.6837
## StorageGood                                               3.0393     1.3327
## StorageUncertain                                          5.0157     1.8418
## Extraction_made_months_after_collection                   0.0170     0.1446
## Extraction_made_years_after_collection:StorageGood        1.5575     2.0179
## Extraction_made_years_after_collection:StorageUncertain   3.1255     2.5851
## StorageGood:Extraction_made_months_after_collection      -0.1166     0.1712
## StorageUncertain:Extraction_made_months_after_collection -0.2706     0.2213
##                                                         z value Pr(>|z|)
## (Intercept)                                              -1.070  0.28448
## Extraction_made_years_after_collection                   -0.205  0.83762
## StorageGood                                               2.281  0.02257 *
## StorageUncertain                                          2.723  0.00646 **
## Extraction_made_months_after_collection                   0.118  0.90643
## Extraction_made_years_after_collection:StorageGood        0.772  0.44023
## Extraction_made_years_after_collection:StorageUncertain   1.209  0.22665
## StorageGood:Extraction_made_months_after_collection      -0.681  0.49589
## StorageUncertain:Extraction_made_months_after_collection -1.223  0.22139
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##               (Intr) Extrctn_md_y__ StrgGd StrgUn Extrctn_md_m__ E____:SG
## Extrctn_md_y__  0.933
## StorageGood    -0.906 -0.848
## StorgUncrtn    -0.675 -0.620          0.637
## Extrctn_md_m__ -0.948 -0.996          0.863  0.631
## Extr____:SG    -0.772 -0.827          0.848  0.498  0.824
## Extr____:SU    -0.614 -0.656          0.564  0.865  0.653          0.540
## StrgG:E____     0.794  0.833         -0.882 -0.515 -0.837         -0.993
## StrgU:E____     0.626  0.656         -0.577 -0.885 -0.658         -0.539
##               E____:SU SG:E__
## Extrctn_md_y__
## StorageGood
## StorgUncrtn
## Extrctn_md_m__
## Extr____:SG
## Extr____:SU
## StrgG:E____     -0.544
## StrgU:E____     -0.997    0.548
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.0749041 (tol = 0.002, component 1)
```

```
#test assumptions for glmer
testDispersion(full)
```
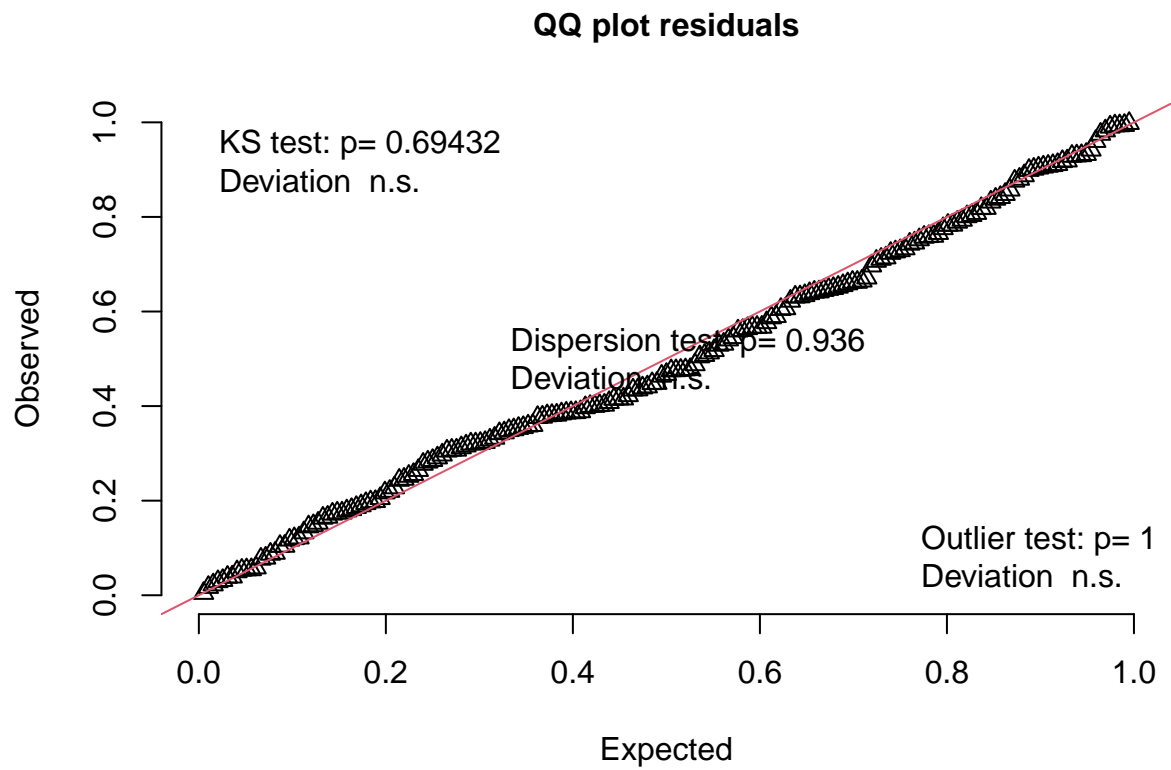
DHARMa nonparametric dispersion test via sd of residuals fitted vs. simulated
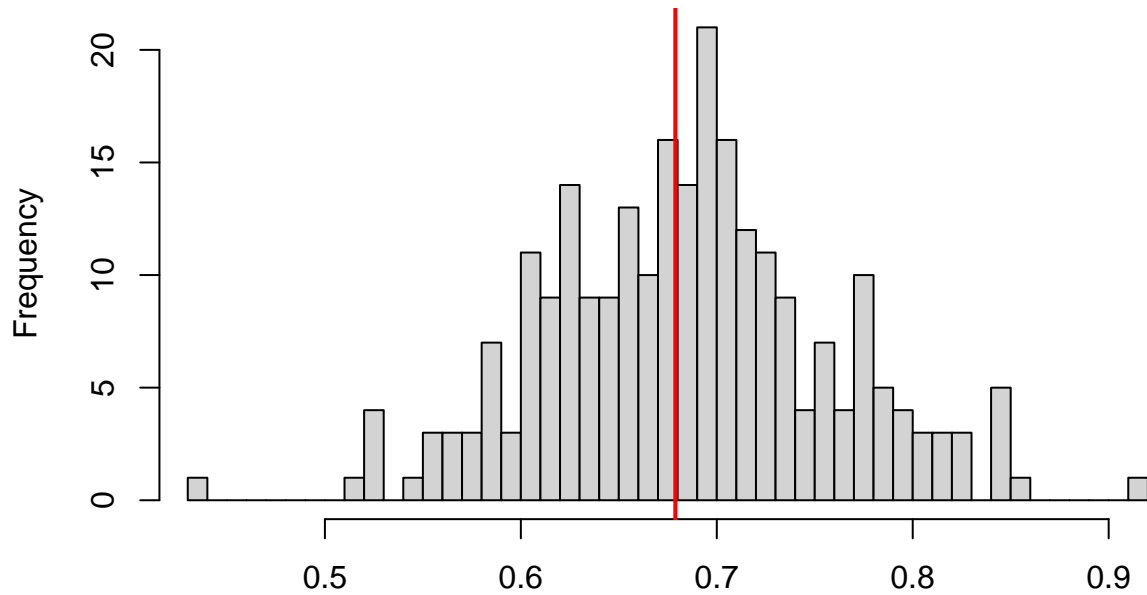
Simulated values, red line = fitted model. p–value (two.sided) = 0.936

```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 0.99186, p-value = 0.936
## alternative hypothesis: two.sided
```

```
testUniformity(full)
```

QQ plot residuals

KS test: p= 0.69432
Deviation n.s.

Dispersion test: p= 0.936
Deviation n.s.

Outlier test: p= 1
Deviation n.s.

Observed

Expected

```
## 
##   One-sample Kolmogorov-Smirnov test
## 
## data:  simulationOutput$scaledResiduals
## D = 0.050853, p-value = 0.6943
## alternative hypothesis: two-sided
```

```
testOutliers(full)
```

Outlier test n.s. **Histogram of frequBoot**

Residuals (outliers are marked red) frequBoot
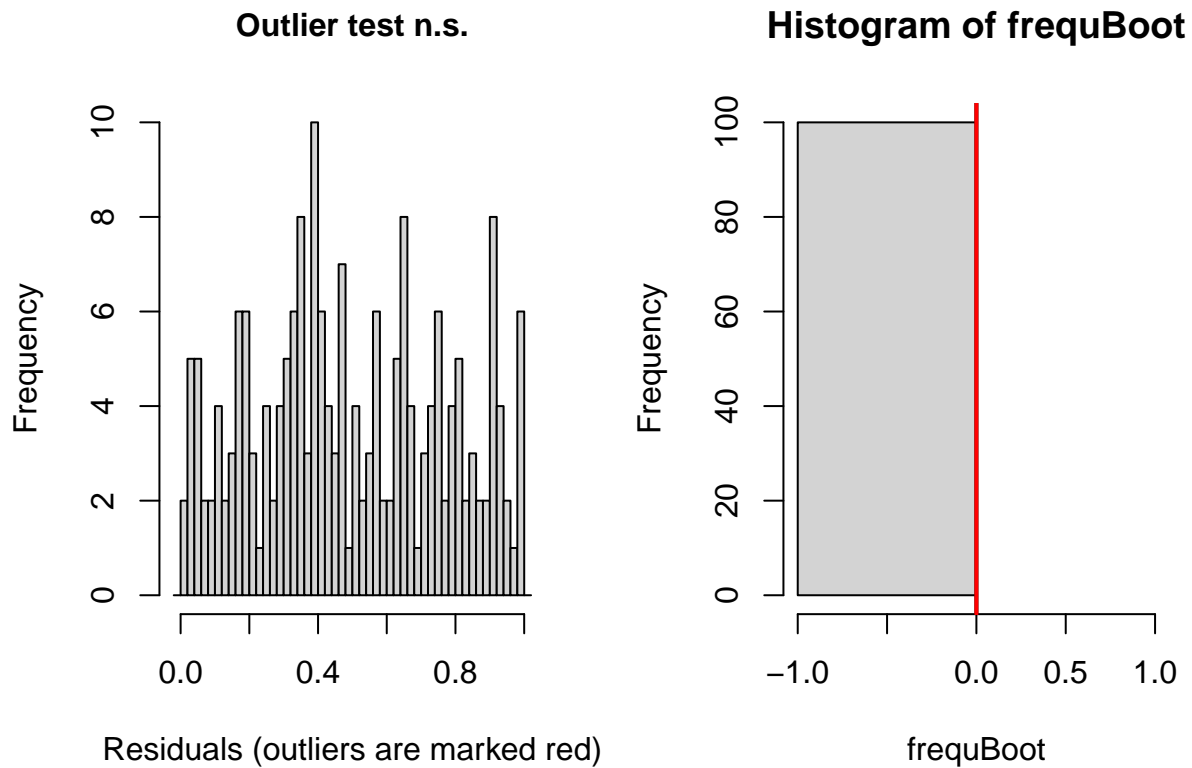
```
## 
##   DHARMa bootstrapped outlier test
## 
## data:  full
## outliers at both margin(s) = 0, observations = 195, p-value = 1
## alternative hypothesis: two.sided
##  percent confidence interval:
##  0 0
## sample estimates:
## outlier frequency (expected: 0 )
##                                  0
```

```
simulationOutput <- simulateResiduals(fittedModel = full, plot = F)
residuals(simulationOutput, quantileFunction = qnorm, outlierValues = c(-7,7))
```

```
##    [1]  1.15439830  0.56387069 -0.25430580 -0.13164567  2.44278269 -0.89362107
##    [7] -1.02591289 -1.64628491 -0.91021718  0.40888704 -2.64184088 -1.34704380
##   [13] -0.31590027  0.31072448 -0.29160370 -1.86089039 -0.37260951 -0.85023023
##   [19] -0.84771516 -0.16493534 -1.77274000 -0.03099095  0.44279474 -0.65724656
##   [25]  2.33210883 -0.50995385  0.65141206  0.16682260 -0.66889381 -2.12617134
##   [31] -1.57151539  1.32480148 -0.93827636 -0.59138002 -0.05680065  1.02668792
##   [37]  0.01088275  0.96035846  0.60928694 -0.46403728 -0.82519099 -0.44463610
##   [43]  1.34646308  0.26908340 -0.09173323  2.10043957 -1.60546634 -1.04814268
##   [49] -0.50425621 -0.30267808 -1.17622694  2.45459622 -0.38492371  1.06691839
##   [55]  0.41781780  0.42518756  0.22628019 -1.26035868  1.21189454 -0.26175675
```

```
## [61]  0.26524861  0.37121981  2.49640339  0.07250264 -0.98167876  1.49329883
## [67] -0.46750900  0.90893974  0.19321752  0.60147411 -0.05788657 -1.35981801
## [73] -0.28864915 -0.36359484 -0.15448218 -0.30432260 -0.97338181 -0.21407998
## [79] -0.25476055  1.38666150  1.39569296  0.16108258 -1.76526985  0.41779114
## [85]  0.51434443  0.65869250  0.90712280 -0.69573209 -0.36055087 -0.37711196
## [91]  0.03170635 -0.19622404 -0.24988785 -0.06057731  0.78037889  0.80320108
## [97]  1.27381444  1.28831872 -0.63079941 -2.01283506  1.33326464  0.59229569
## [103]  0.81745683  1.31458229 -0.88019305  0.85305032  0.37732350 -0.56758953
## [109] -0.78364955 -1.89624886 -0.31388942 -1.26207363  0.15128630 -1.60753260
## [115] -0.50467740 -0.06288995 -0.47919258  0.38975742  1.56760120  0.01811059
## [121]  0.07792022  0.34183387 -0.16581301 -0.74750972 -0.06250150  1.17445380
## [127]  0.62121719  0.35309725  0.70198346  1.47451628  0.86920572 -1.42920875
## [133]  0.98494620  0.10075594  0.54278594 -1.11624782  0.36104741 -1.05895317
## [139]  0.16800947 -0.57751432 -0.93838281 -0.28396583 -1.61538696  0.36308454
## [145] -0.94721399  2.85674687 -0.92087049 -0.14891616  0.84402738  0.39897050
## [151] -0.45880637 -1.19061776 -0.43371804 -1.16798544  0.78187143 -0.77240284
## [157]  0.69726444  1.46994379  0.72337193  0.99970245 -0.55163818  0.55521294
## [163]  0.70414058 -0.86335496 -0.30626038  1.46403451  1.96004787 -0.40742683
## [169]  1.30230328 -0.08241480  0.11311577  0.33358500 -1.41382077  0.21935008
## [175]  0.14859586 -0.21579861  0.38222060 -0.38962792  0.75672019  0.33514309
## [181]  0.68050416 -0.53386626 -0.46712736 -0.28907534 -0.68871319 -0.23327001
## [187] -0.13018282  0.16855451  1.74814612 -0.40261792 -0.21425613 -0.29579414
## [193] -0.24777170  0.04417442 -0.48997372
```

```
plot(simulationOutput)
```

DHARMa residual

QQ plot residuals

KS test: p= 0.694...
Deviation n.s.

Dispersion test: p= 0.93...
Deviation n.s.

Outlier test: p= 1
Deviation n.s.

Observed / Expected

Residual vs. predicted
No significant problems detected

DHARMa residual / Model predictions (rank transformed)