

```
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()
```

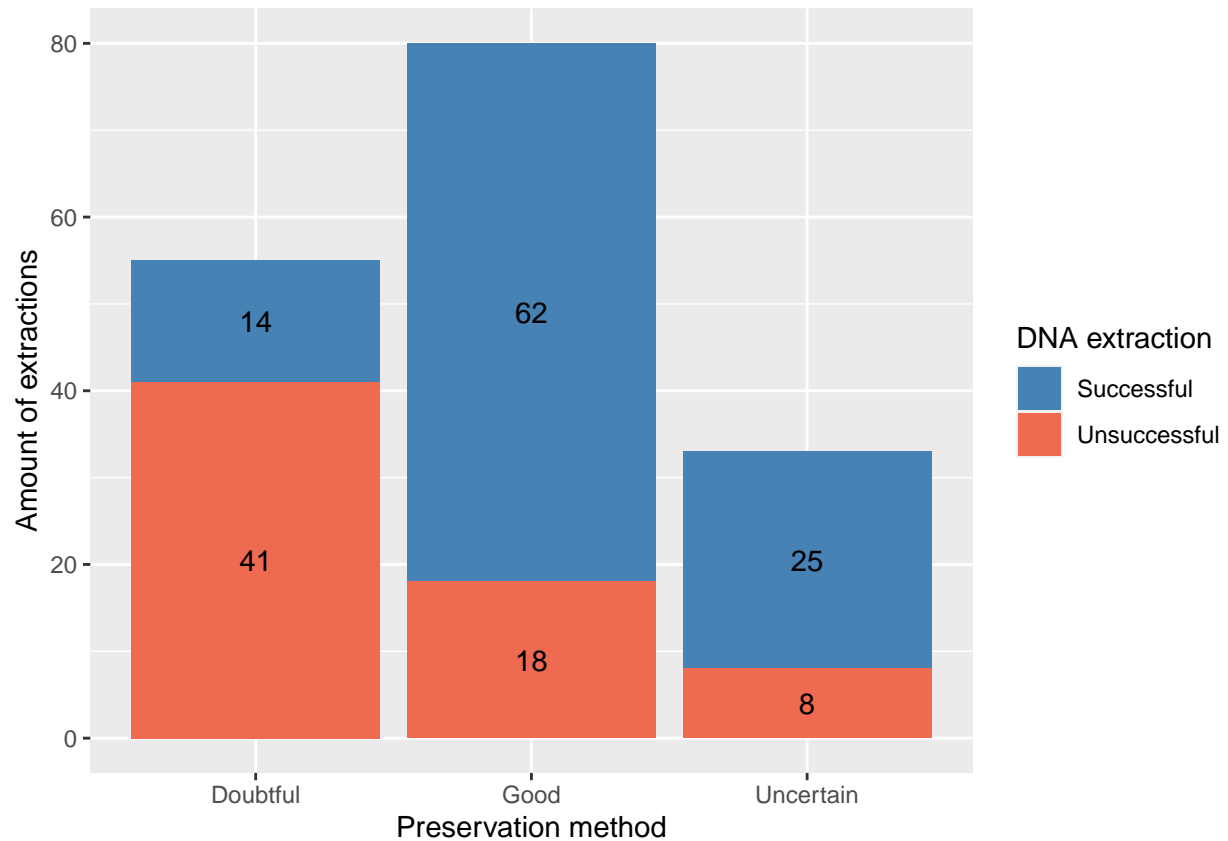
```
## x purrr::set_names()   masks magrittr::set_names()
```

```
## x tidyr::unpack()      masks Matrix::unpack()
```

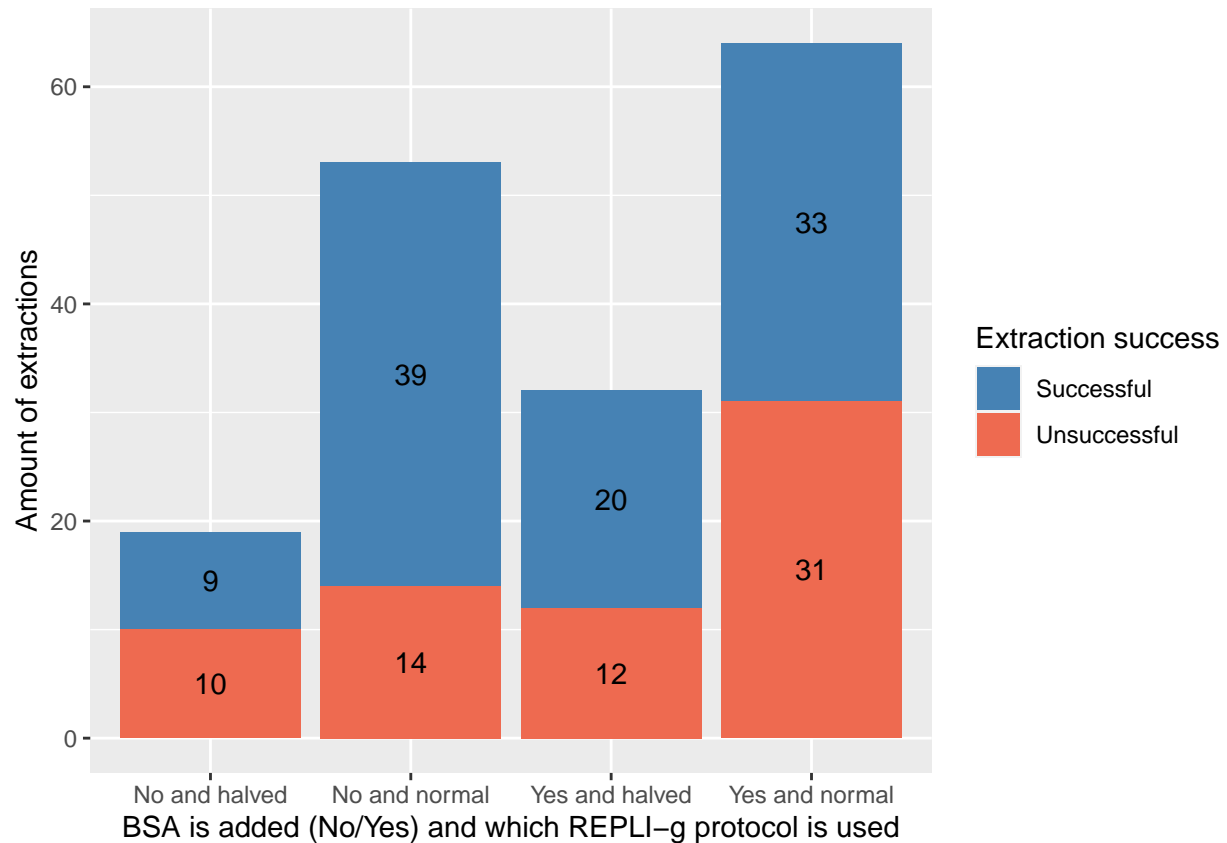
```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
DNA_extractie_protocollen <- read_excel("F:/UGent/2022-2023/Masterthesis/DNA extractie protocollen.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'))
```



```
DNA_extractie_protocol %>% #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 ***
```

```
## 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 = "response")
```

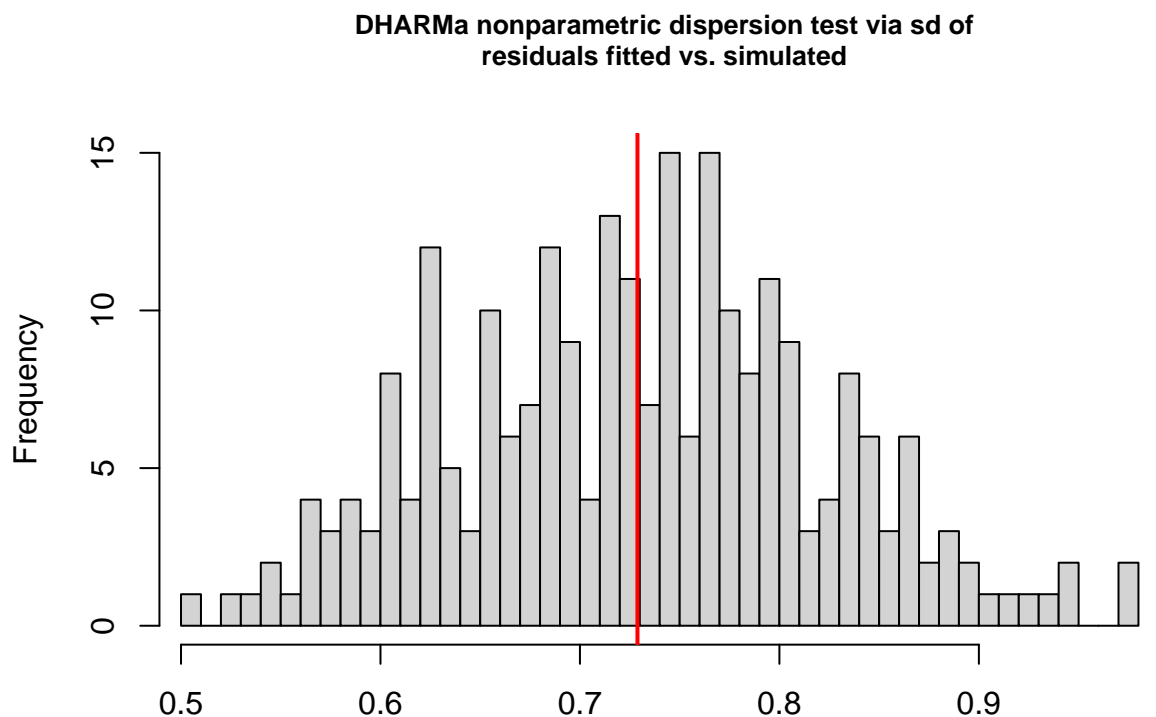
```
## $emmeans
##      Storage      prob      SE  df asymp.LCL asymp.UCL
## Doubtful  0.167 0.0731 Inf    0.0669    0.360
## Good      0.824 0.0657 Inf    0.6581    0.919
## Uncertain 0.873 0.0843 Inf    0.6076    0.969
##
## Results are averaged over the levels of: BSA_added, RepliG
## Confidence level 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 = "response")
```

```
## $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
```

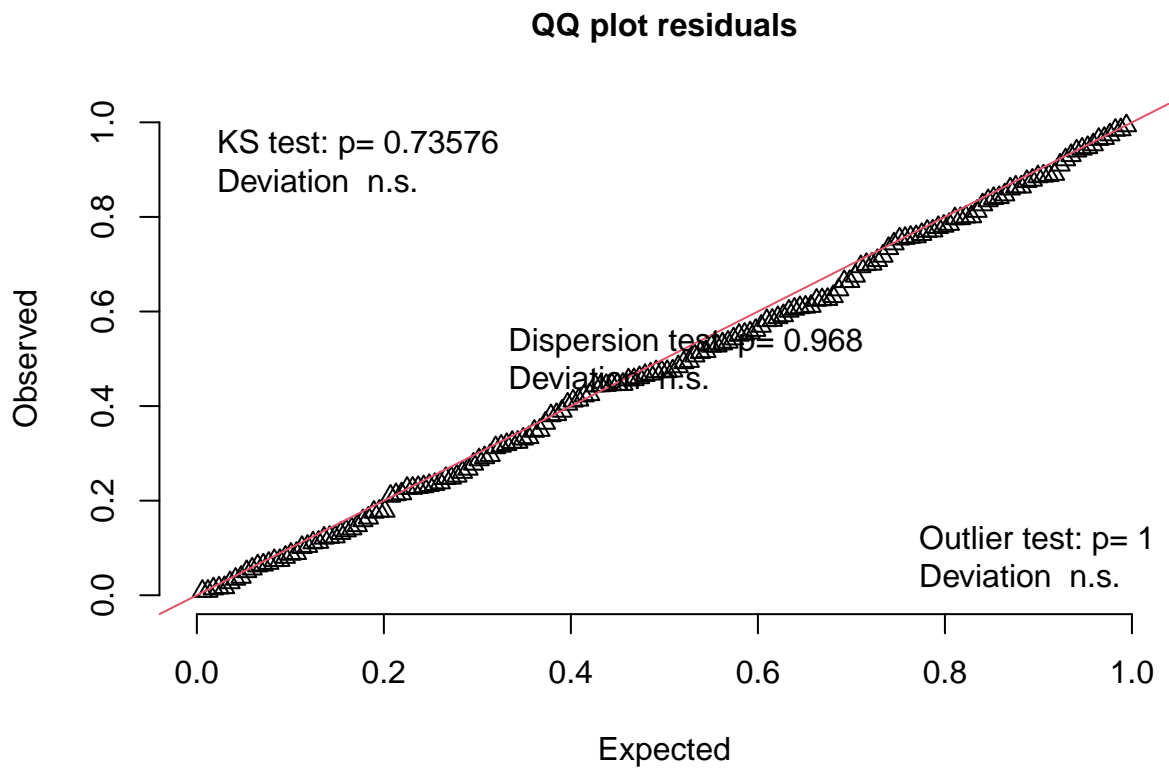
```
## No halved / Yes halved      0.0742 0.0719 Inf    1 -2.685 0.0365
## No halved / No normal      0.1758 0.1326 Inf    1 -2.305 0.0969
## No halved / Yes normal     0.2128 0.1666 Inf    1 -1.977 0.1968
## Yes halved / No normal     2.3679 1.7910 Inf    1  1.140 0.6649
## Yes halved / Yes normal    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)
```



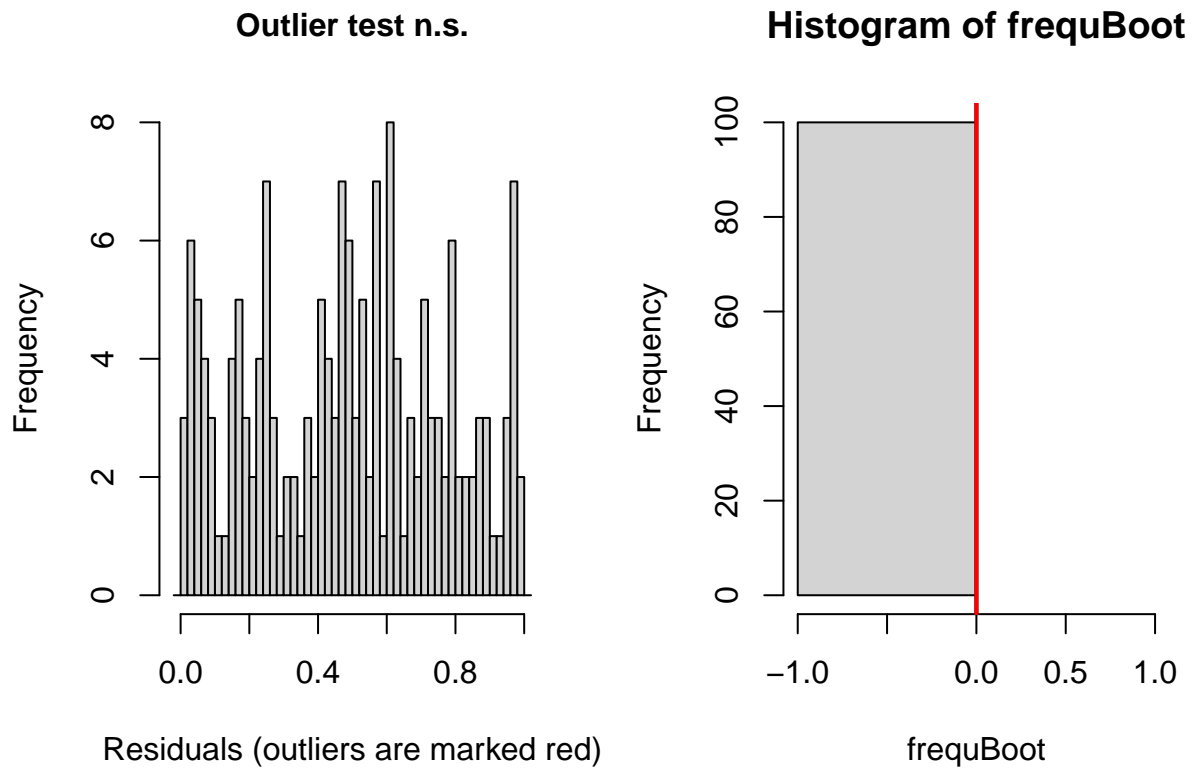
```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data:  simulationOutput
## dispersion = 0.99935, p-value = 0.968
## alternative hypothesis: two.sided
```

```
testUniformity(full)
```



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

```
testOutliers(full)
```



```
##
## 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))
```

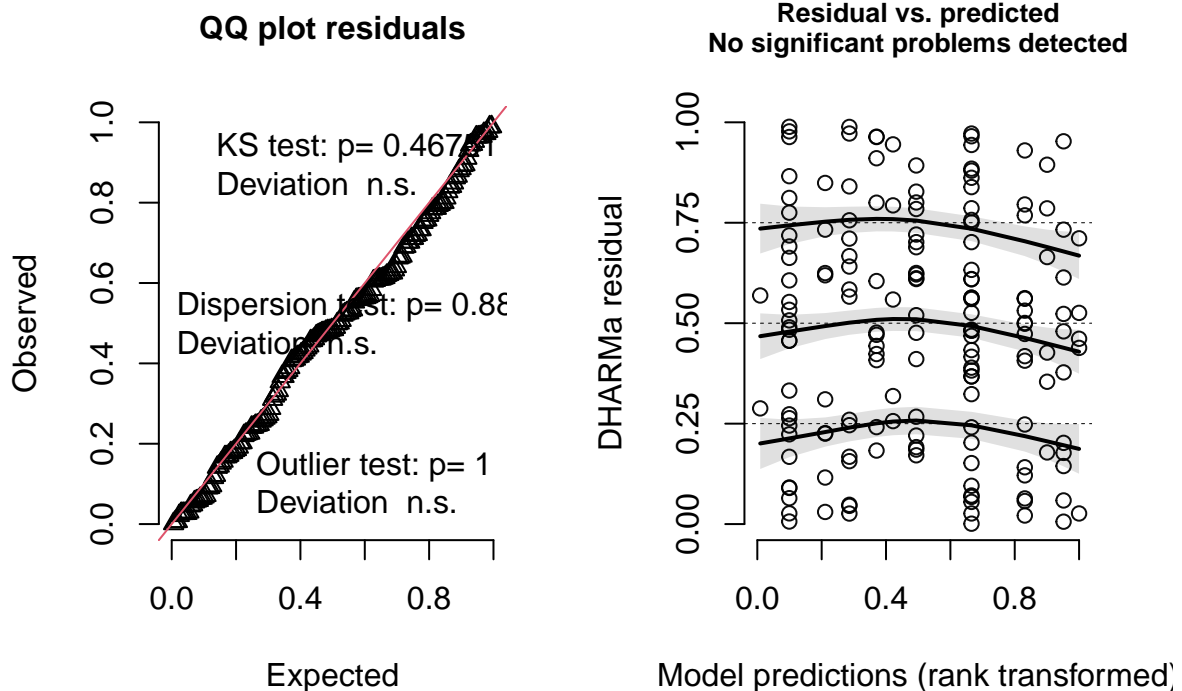
```
## [1] -0.079292231 1.585778168 -0.703442924 0.696958721 -0.832258911
## [6] -1.467321537 0.425023922 -0.232691378 -1.935906179 1.801223173
## [11] -1.596387202 -0.702051910 -0.058563047 -0.458152559 1.173183523
## [16] -0.904070489 0.151483212 -1.308470296 -1.493509965 1.790737248
## [21] -3.092608948 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
```

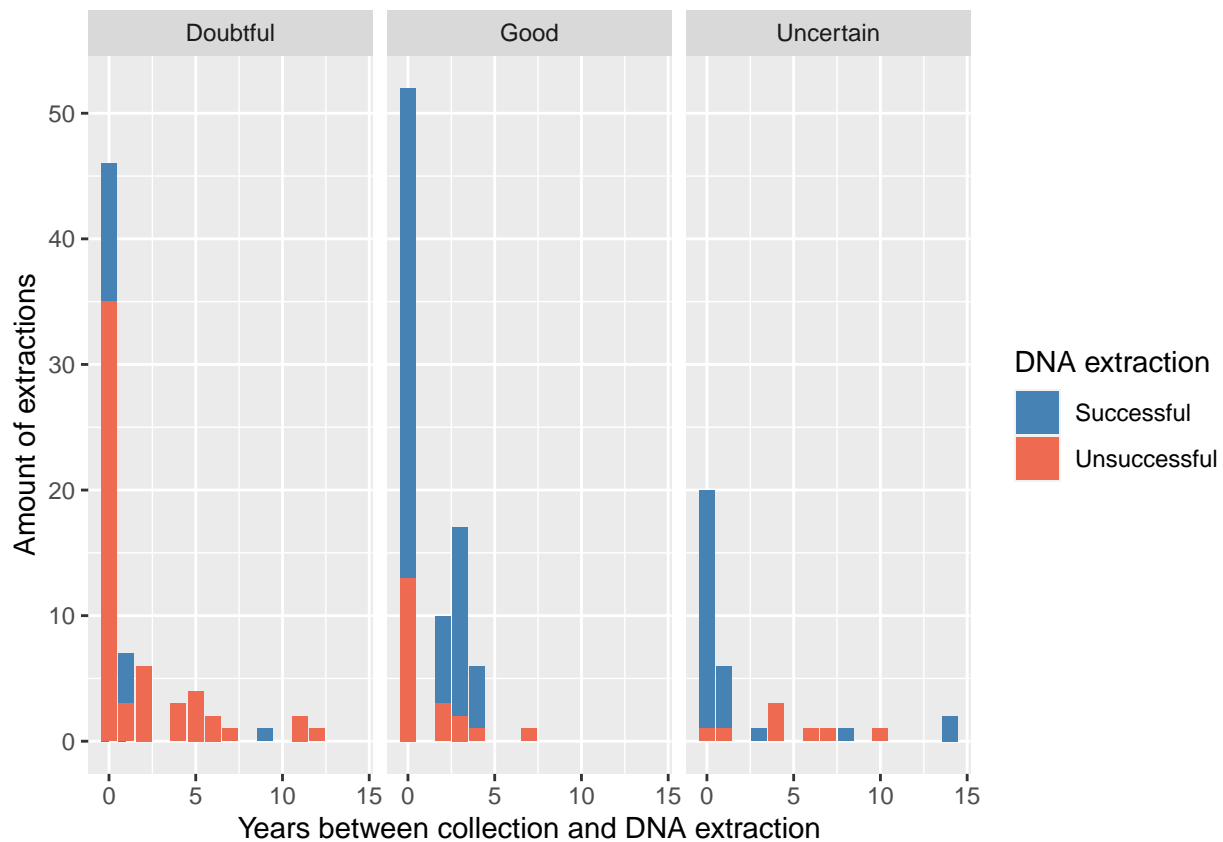
```
plot(simulationOutput)
```

DHARMA residual



```
datum_over_jaar <- read_excel("F:/UGent/2022-2023/Masterthesis/DNA extractie protocollen over maanden.xlsx")
```

```
datum_over_jaar %>% #Barplots of extractions success depending on the years between collection and DNA
  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'))
```



```
datum_over_jaar %>% #Barplots of extractions success depending on the months between collection and DNA
  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'))
```

