

Microbial persistence in agricultural soil - code/output

Naim Montazeri and Yusuf Nuradeen Garba

2025-08-17

Table of contents

1. Packages	2
2. Reading persistence data and initial setup.....	2
3. T1D	4
3.1 Model Functions	4
3.2 <i>E. coli</i>	5
3.2.1 Fitting models.....	5
3.2.2 Model performance and inference.....	8
3.2.3 Assess covariance/collinearity.....	15
3.2.4 Plot microbial inactivation	16
3.3 <i>Tulane virus TCID50</i>	18
3.3.1 Fitting models.....	18
3.3.2 Model performance and inference.....	20
3.3.3 Assess covariance/collinearity.....	28
3.3.4 Plot microbial inactivation	30
3.4 Combined <i>E. coli</i> and Tulane virus plot.....	31
3.5 TuV and HuNoV - <i>RNase RT-qPCR</i>	33
3.5.1 Fitting models.....	33
3.5.2 Model performance and inferences.....	35
3.5.3 Assess covariance/collinearity.....	46
3.5.4 Plot microbial inactivation	49
4. T2D	52
4.1 Model Functions	52
4.2 <i>E. coli</i>	52
4.3 <i>Tulane virus TCID50</i>	54
4.4 TuV and HuNoV - <i>RNase RT-qPCR</i>	57
5. T3D	60
5.1 Model Functions	60

5.2 <i>E. coli</i>	61
5.3 <i>Tulane virus TCID50</i>	63
5.4 TuV and HuNoV - <i>RNase RT-qPCR</i>	65
6. Combined TD values.....	69
7. Correlation matrix.....	72
8. PCR:Inf ratio for Tulane virus.....	75
9. Convert TCID50 to PFU.....	80
10. Weather/soil conditions.....	81
10.1 Soil moisture changes.....	81
10.2 Monthly soil temperature.....	82
10.3 Seven days prior to sample collection.....	85

1. Packages

Run this script from within the R project - the code now uses relative paths which may not work otherwise.

```
# Load packages
library(emmeans)
library(GGally)
library(ggpubr)
library(ggResidpanel)
library(here)
library(marginaleffects)
library(MASS)
library(multcomp)
library(multcompView)
library(nlme)
library(performance)
library(tidyverse)

# Register system fonts in R, if necessary
## library(extrafont)
## font_import()

# update.packages(ask = FALSE)
```

2. Reading persistence data and initial setup

```
readRDS(here('obj2', 'data', 'obj2_dat.RDS')) |>
  str()
```

```
'data.frame': 480 obs. of 8 variables:
 $ microbe : Factor w/ 3 levels "ecoli","nv","tv": 1 1 1 1 1 1 1 1 1 1 ...
 $ matrix : Factor w/ 2 levels "ps","s": 1 1 1 1 1 1 1 1 1 1 ...
 $ temp : Factor w/ 1 level "12": 1 1 1 1 1 1 1 1 1 1 ...
 $ day : num 0 0 0 1 1 1 3 3 3 7 ...
 $ replicate: num 1 2 3 1 2 3 1 2 3 1 ...
 $ assay : Factor w/ 3 levels "pcr","plate",...: 2 2 2 2 2 2 2 2 2 2 ...
 $ survival : num 6.78 6.59 6.8 7.21 7.15 ...
 $ week : num 0 0 0 0.143 0.143 ...
```

```
# Limits of detection
```

```
LOD <- data.frame(
  microbe = as.factor(c('ecoli_plate', 'tv', 'tv_tcid50', 'nv')),
  lod = c(1.38, 3.78, 0.5, 2.15)
)
```

```
# iterations
```

```
ITER <- 5000
```

```
# Set random number generation state
```

```
SEED <- 5638
```

```
# Limit the number of significant digits to 3
```

```
options(digits=3)
```

```
# Turn in environments/packages/versions/etc
```

```
sessioninfo::session_info (pkgs = c("attached"))
```

```
– Session info
```

```
setting value
version R version 4.5.0 (2025-04-11)
os macOS Sequoia 15.6
system aarch64, darwin20
ui X11
language (EN)
collate en_US.UTF-8
ctype en_US.UTF-8
tz America/New_York
date 2025-08-17
pandoc 3.4 @
/Applications/RStudio.app/Contents/Resources/app/quarto/bin/tools/aarch64/
(via rmarkdown)
quarto 1.3.433 @ /usr/local/bin/quarto
```

```
– Packages
```

```
package * version date (UTC) lib source
dplyr * 1.1.4 2023-11-17 [1] CRAN (R 4.5.0)
emmeans * 1.11.2 2025-07-11 [1] CRAN (R 4.5.0)
```

forcats	* 1.0.0	2023-01-29	[1]	CRAN	(R 4.5.0)
GGally	* 2.3.0	2025-07-18	[1]	CRAN	(R 4.5.0)
ggplot2	* 3.5.2	2025-04-09	[1]	CRAN	(R 4.5.0)
ggpubr	* 0.6.1	2025-06-27	[1]	CRAN	(R 4.5.0)
ggResidpanel	* 0.3.0	2019-05-31	[1]	CRAN	(R 4.5.0)
here	* 1.0.1	2020-12-13	[1]	CRAN	(R 4.5.0)
lubridate	* 1.9.4	2024-12-08	[1]	CRAN	(R 4.5.0)
marginalEffects	* 0.28.0	2025-06-25	[1]	CRAN	(R 4.5.0)
MASS	* 7.3-65	2025-02-28	[1]	CRAN	(R 4.5.0)
multcomp	* 1.4-28	2025-01-29	[1]	CRAN	(R 4.5.0)
multcompView	* 0.1-10	2024-03-08	[1]	CRAN	(R 4.5.0)
mvtnorm	* 1.3-3	2025-01-10	[1]	CRAN	(R 4.5.0)
nlme	* 3.1-168	2025-03-31	[1]	CRAN	(R 4.5.0)
performance	* 0.15.0	2025-07-10	[1]	CRAN	(R 4.5.0)
purrr	* 1.1.0	2025-07-10	[1]	CRAN	(R 4.5.0)
readr	* 2.1.5	2024-01-10	[1]	CRAN	(R 4.5.0)
stringr	* 1.5.1	2023-11-14	[1]	CRAN	(R 4.5.0)
survival	* 3.8-3	2024-12-17	[1]	CRAN	(R 4.5.0)
TH.data	* 1.1-3	2025-01-17	[1]	CRAN	(R 4.5.0)
tibble	* 3.3.0	2025-06-08	[1]	CRAN	(R 4.5.0)
tidyr	* 1.3.1	2024-01-24	[1]	CRAN	(R 4.5.0)
tidyverse	* 2.0.0	2023-02-22	[1]	CRAN	(R 4.5.0)

[1] /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/library

* — Packages attached to the search path.

3. T1D

3.1 Model Functions

Log linear and non-linear models

Log-linear and reparameterized log-linear

```
ll <- function(x, log10n0, alpha) {
  log10n0 - (x*alpha)
}
```

```
llrepar <- function(x, log10n0, TD, D = 1){
  log10n0 - D*x/TD
}
```

Nonlinear models

Weibull and reparameterized Weibull

```
wb <- function(x, log10n0, alpha, beta) {
  log10n0 - (x/exp(alpha))^beta
}
```

```

wbrepar <- function(x, log10n0, TD, beta, D = 1){
  log10n0 - D*(x/TD)^beta
}

# Log-logistic and reparameterized log-logistic
lg = function(x, log10n0, kappa, sigma){
  log10n0 - log10(1+ exp((log(x)-kappa)/sigma^2))
}

lgrepar = function(x, log10n0, TD, sigma, D = 1){
  log10n0 - log10(1+ exp((log(x)- (log(TD) - log(10^D -
1)*sigma^2))/sigma^2))
}

# To clean the environment
list_exc <- c(lsf.str(envir = .GlobalEnv), "LOD", "ITER", "SEED", "list_exc",
"ec_m2")

```

3.2 *E. coli*

3.2.1 Fitting models

```

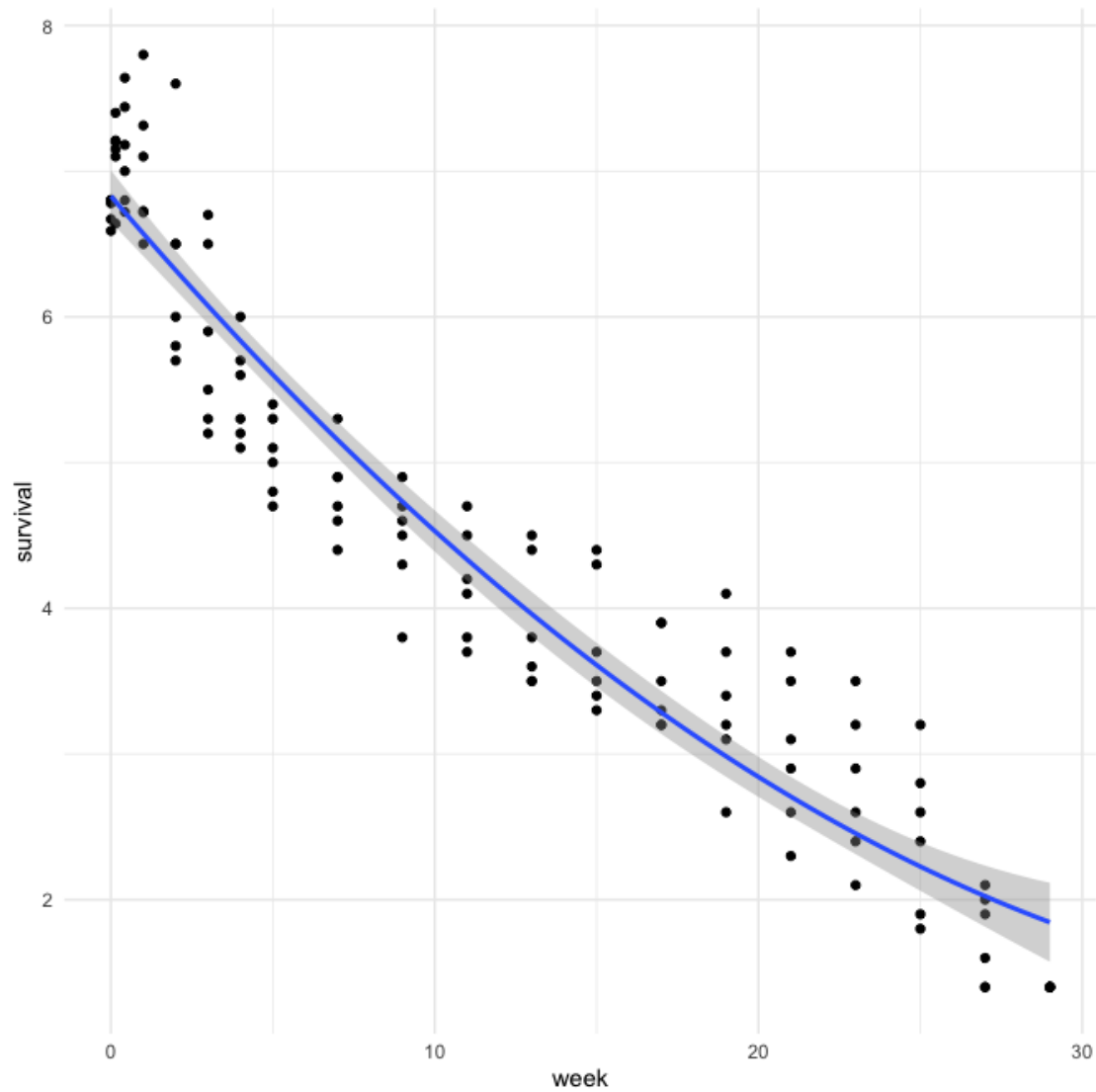
m2 <- readRDS(here('obj2', 'data', 'obj2_dat.RDS')) |>
  dplyr::filter(microbe == 'ecoli',
                assay == "plate") |>
  droplevels()

# Levels
contrasts(m2$matrix)

      s
ps 0
s   1

# Initial visual inspection
ggplot(m2, aes(x = week, y = survival)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
  theme_minimal()

```



```

m2_mod_null = nlme::gnls(survival ~ ll(x = week,
                                     log10n0 = log10n0,
                                     alpha = alpha),
                         data = m2,
                         params = log10n0+alpha ~ 1,
                         start = list(log10n0 = 5,
                                     alpha = 1))

m2_mod_ll = nlme::gnls(survival ~ ll(x = week,
                                     log10n0 = log10n0,
                                     alpha = alpha),
                      data = m2,
                      params = log10n0+alpha ~ matrix,
                      start = list(log10n0 = c(5,0),
                                  alpha = c(1,0)))

```

```

m2_mod_llrepar = nlme::gnls(survival ~ llrepar(x = week,
                                             log10n0 = log10n0,
                                             TD = TD),
                           data = m2,
                           params = log10n0+TD ~ matrix,
                           start = list(log10n0 = c(6, 0),
                                         TD = c(5, 0)))

m2_mod_wb = nlme::gnls(survival ~ wb(x = week,
                                     log10n0 = log10n0,
                                     alpha = alpha,
                                     beta = beta),
                      data = m2,
                      params = log10n0+alpha+beta ~ matrix,
                      start = list(log10n0 = c(5, 0),
                                   alpha = c(1, 0),
                                   beta = c(1, 0)))

m2_mod_wbrepar = nlme::gnls(survival ~ wbrepar(x = week,
                                                log10n0 = log10n0,
                                                TD = TD,
                                                beta = beta),
                           data = m2,
                           params = log10n0+TD+beta ~ matrix,
                           start = list(log10n0 = c(5, rep(0, 1)),
                                         TD = c(1, 0),
                                         beta = c(1, 0)))

m2_mod_lg = nlme::gnls(survival ~ lg(x = week,
                                     log10n0 = log10n0,
                                     kappa = kappa,
                                     sigma = sigma),
                      data = m2,
                      params = log10n0+kappa+sigma ~ matrix,
                      start = list(log10n0 = c(5, 0),
                                   kappa = c(2, 0),
                                   sigma = c(1, 0)))

m2_mod_lgpar = nlme::gnls(survival ~ lgrepar(x = week,
                                              log10n0 = log10n0,
                                              sigma = sigma,
                                              TD = TD),
                          data = m2,
                          params = log10n0+sigma+TD ~ matrix,
                          start = list(log10n0 = c(4, 0),
                                        sigma = c(0.5, 0),
                                        TD = c(4, 0)))

```

3.2.2 Model performance and inference

```
# Model comparisons
```

```
performance::compare_performance (
  m2_mod_null, m2_mod_ll, m2_mod_llrepar,
  m2_mod_lg, m2_mod_lgpar, m2_mod_wb, m2_mod_wbrepar,
  metrics = c('AICc'))
```

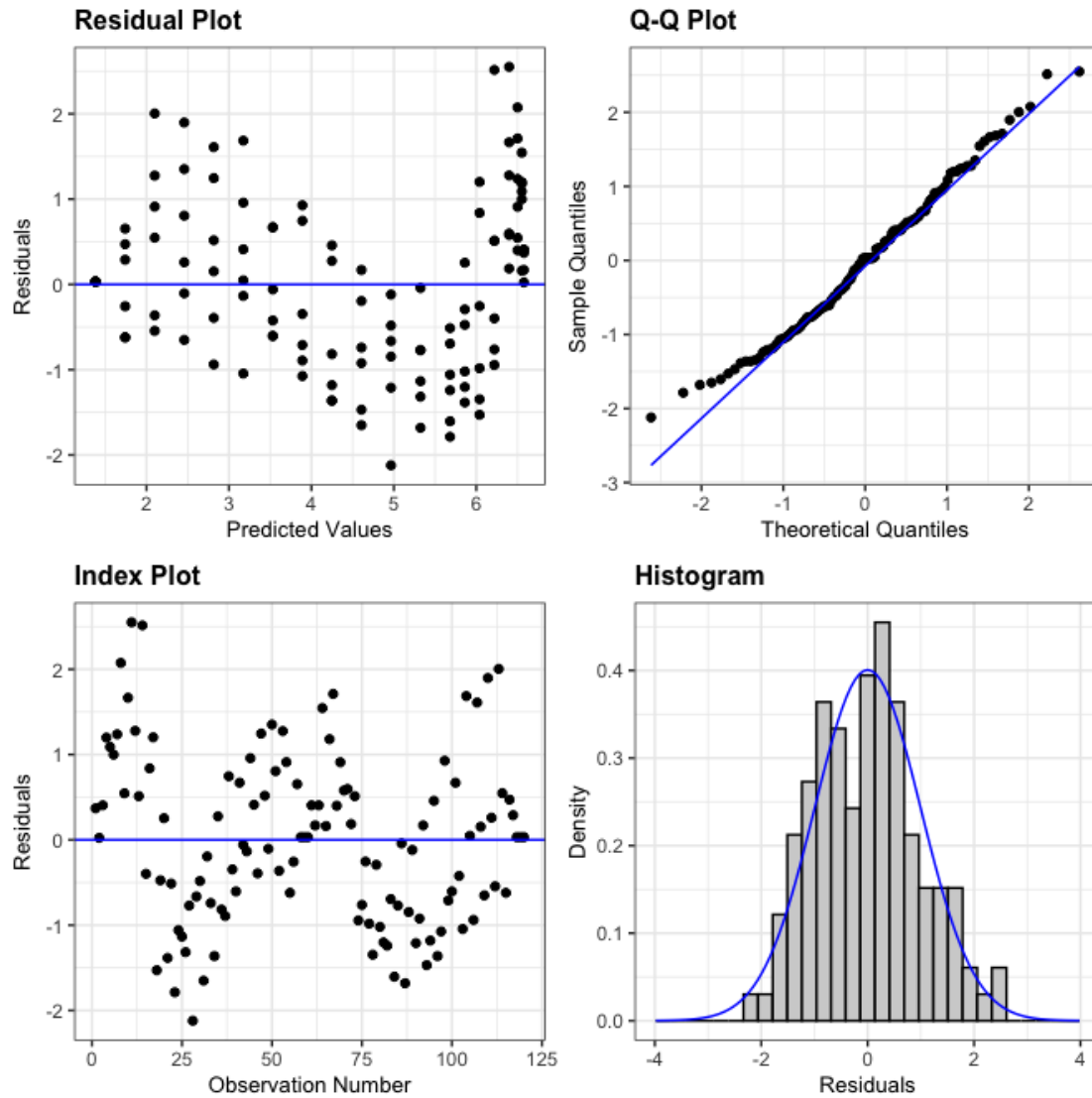
```
# Comparison of Model Performance Indices
```

Name	Model	AIC (weights)	AICc (weights)	BIC (weights)
m2_mod_null	gnls	200.8 (<.001)	201.0 (<.001)	209.1 (<.001)
m2_mod_ll	gnls	202.2 (<.001)	202.7 (<.001)	216.1 (<.001)
m2_mod_llrepar	gnls	202.2 (<.001)	202.7 (<.001)	216.1 (<.001)
m2_mod_lg	gnls	186.9 (<.001)	187.9 (<.001)	206.4 (<.001)
m2_mod_lgpar	gnls	186.9 (<.001)	187.9 (<.001)	206.4 (<.001)
m2_mod_wb	gnls	167.9 (0.500)	168.9 (0.500)	187.4 (0.500)
m2_mod_wbrepar	gnls	167.9 (0.500)	168.9 (0.500)	187.4 (0.500)

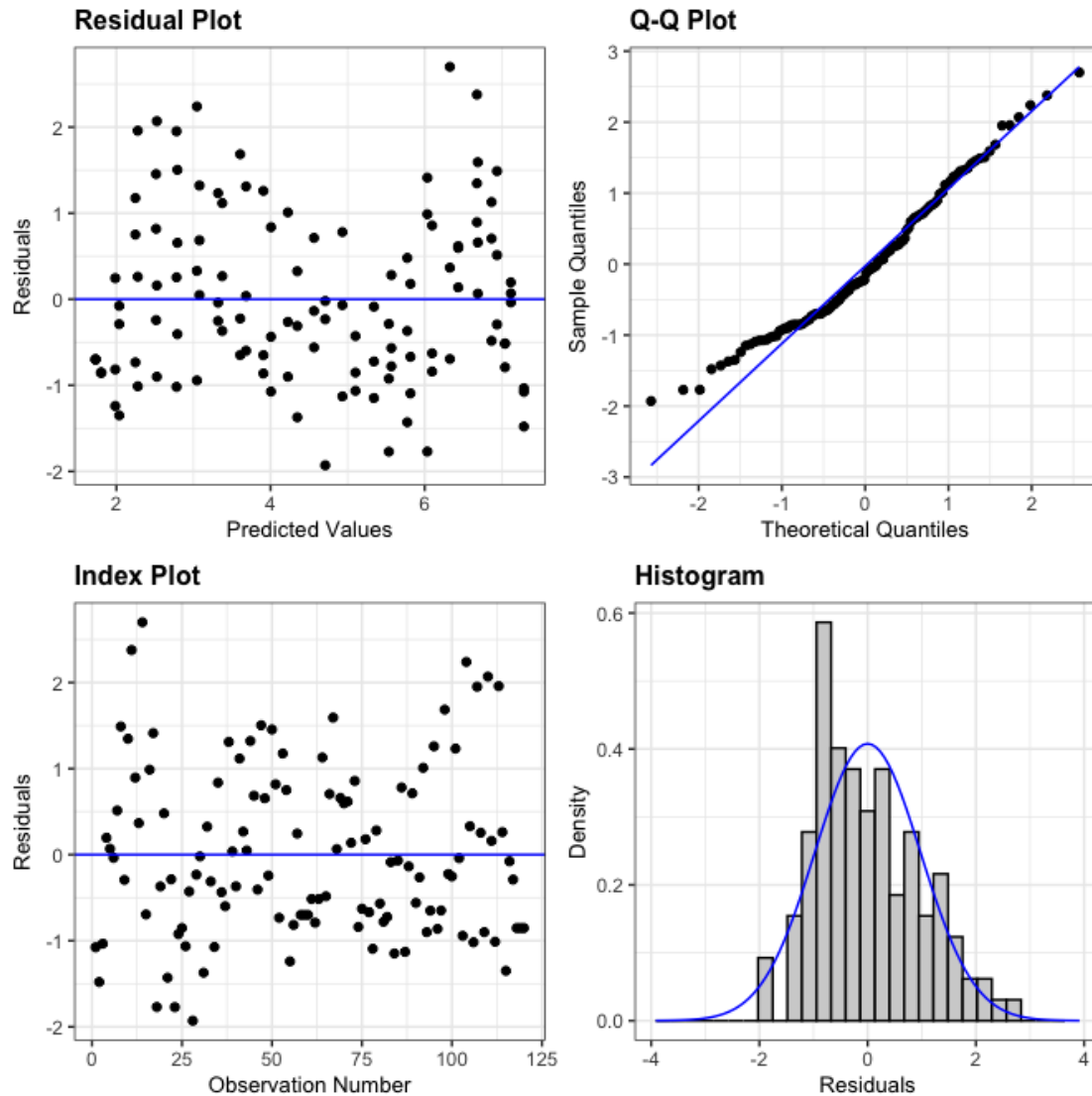
Name	RMSE	Sigma
m2_mod_null	0.545	0.549
m2_mod_ll	0.539	0.548
m2_mod_llrepar	0.539	0.548
m2_mod_lg	0.497	0.510
m2_mod_lgpar	0.497	0.510
m2_mod_wb	0.459	0.471
m2_mod_wbrepar	0.459	0.471

```
# Residual plots
```

```
m2_res_null = resid(m2_mod_null, type = 'normalized')
m2_fit_null = fitted(m2_mod_null)
resid_auxpanel(m2_res_null, m2_fit_null)
```

```
m2_res_wbregar = resid(m2_mod_wbregar, type = 'normalized')
m2_fit_wbregar = fitted(m2_mod_wbregar)
resid_auxpanel(m2_res_wbregar, m2_fit_wbregar)
```



```
pres_mod <- m2_mod_wbrepair
```

```
# gnls model tuning
m2_mod_exp = update(pres_mod, weights = varExp())
m2_mod_exp_wk = update(pres_mod, weights = varExp(form = ~week))
m2_mod_exp_sur = update(pres_mod, weights = varExp(form = ~survival))
m2_mod_idt_mtx = update(pres_mod, weights = varIdent(form = ~1 | matrix))
#m2_mod_idt_mic = update(pres_mod, weights = varIdent(form = ~1 | microbe))
#m2_mod_pow = update(pres_mod, weights = varPower())
m2_mod_pow_sur = update(pres_mod, weights = varPower(form = ~ survival))

#m2_mod_combined1 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
microbe), varExp(form = ~ survival)))
#m2_mod_combined2 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
microbe), varExp(form = ~ week)))
m2_mod_combined3 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
```

```
matrix), varExp(form = ~ survival)))
m2_mod_combined4 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
matrix), varExp(form = ~ week)))
```

```
compare_performance(
  pres_mod,
  m2_mod_exp,
  m2_mod_exp_wk,
  m2_mod_idt_mtx,
  #m2_mod_idt_mic,
  #m2_mod_pow,
  #m2_mod_combined1,
  #m2_mod_combined2,
  m2_mod_combined3,
  m2_mod_combined4,
  metrics = c('AICc'))
```

Comparison of Model Performance Indices

Name	Model	AIC (weights)	AICc (weights)	BIC (weights)
pres_mod	gnls	167.9 (0.354)	168.9 (0.400)	187.4 (0.740)
m2_mod_exp	gnls	169.8 (0.132)	171.1 (0.129)	192.1 (0.068)
m2_mod_exp_wk	gnls	169.9 (0.131)	171.2 (0.127)	192.2 (0.068)
m2_mod_idt_mtx	gnls	169.1 (0.191)	170.4 (0.186)	191.4 (0.099)
m2_mod_combined3	gnls	170.0 (0.120)	171.7 (0.099)	195.1 (0.015)
m2_mod_combined4	gnls	171.1 (0.072)	172.7 (0.059)	196.1 (0.009)

Name	R2	RMSE	Sigma
pres_mod	0.935	0.459	0.471
m2_mod_exp	0.935	0.459	0.459
m2_mod_exp_wk	0.935	0.459	0.473
m2_mod_idt_mtx	0.935	0.459	0.497
m2_mod_combined3	0.935	0.460	0.402
m2_mod_combined4	0.935	0.459	0.490

```
m2_mod <- pres_mod
```

```
summary (m2_mod)
```

Generalized nonlinear least squares fit

Model: survival ~ wbrepar(x = week, log10n0 = log10n0, TD = TD, beta = beta)

Data: m2

AIC BIC logLik

168 187 -76.9

Coefficients:

Value Std.Error t-value p-value

log10n0.(Intercept)	7.29	0.181	40.2	0.0000
log10n0.matrixs	-0.24	0.259	-0.9	0.3460
TD.(Intercept)	2.13	0.567	3.8	0.0003
TD.matrixs	0.05	0.838	0.1	0.9511
beta.(Intercept)	0.66	0.062	10.6	0.0000
beta.matrixs	-0.02	0.089	-0.2	0.8521

Correlation:

	l100.(lg100.	TD.(I)	TD.mtr	bt.(I)
log10n0.matrixs	-0.701				
TD.(Intercept)	-0.876	0.615			
TD.matrixs	0.593	-0.878	-0.676		
beta.(Intercept)	-0.792	0.556	0.981	-0.664	
beta.matrixs	0.549	-0.794	-0.680	0.980	-0.694

Standardized residuals:

Min	Q1	Med	Q3	Max
-1.930	-0.745	-0.180	0.690	2.700

Residual standard error: 0.471

Degrees of freedom: 120 total; 114 residual

```
saveRDS(m2_mod, here('obj2', 'results', 'ecoli_mod_sum.RDS'))
```

```
# Marginal means (predictions)
```

```
m2_mod <- readRDS(here('obj2', 'results', 'ecoli_mod_sum.RDS'))
```

```
expand.grid(
```

```
  matrix = levels(m2$matrix),
  microbe = levels(m2$microbe),
  assay = levels(m2$assay),
  week = seq(0, max(m2$week), 0.5)) |>
```

```
marginalEffects::avg_predictions(
```

```
  model=m2_mod,
  conf_level = 0.95,
  by = c('matrix', 'week')) |>
```

```
saveRDS(here('obj2', 'results', 'ecoli_mod_preds.RDS'))
```

```
# Extract parameters (coefficients) and their 95% CI
```

```
m2_mod_coef = coef(m2_mod)
```

```
m2_mod_confint = confint(m2_mod)
```

```
(mod_params <- as.data.frame (cbind(
```

```
  Param = names (m2_mod_coef),
  Est = round (m2_mod_coef, 3),
  lwr = round (m2_mod_confint[,1], 3),
  upr = round (m2_mod_confint[,2], 3)
)))
```

	Param	Est	lwr	upr
log10n0.(Intercept)	log10n0.(Intercept)	7.287	6.932	7.643
log10n0.matrixs	log10n0.matrixs	-0.245	-0.752	0.262
TD.(Intercept)	TD.(Intercept)	2.127	1.017	3.238
TD.matrixs	TD.matrixs	0.051	-1.591	1.694
beta.(Intercept)	beta.(Intercept)	0.657	0.536	0.778
beta.matrixs	beta.matrixs	-0.017	-0.191	0.158

```

# Estimated marginal (group level) means
list(
  log10n0 = joint_tests (m2_mod, param = 'log10n0'),
  TD = joint_tests (m2_mod, param = 'TD'),
  beta = joint_tests (m2_mod, param = 'beta'))

$log10n0
  model term df1 df2 F.ratio p.value
  matrix      1 113   0.896  0.3460

$TD
  model term df1 df2 F.ratio p.value
  matrix      1 113   0.004  0.9510

$beta
  model term df1 df2 F.ratio p.value
  matrix      1 113   0.035  0.8520

# Estimated marginal (group level) means and contrasts
list(
  log10n0 = emmeans (m2_mod, pairwise ~ matrix, param = 'log10n0'),
  TD = emmeans (m2_mod, pairwise ~ matrix, param = 'TD'),
  beta = emmeans (m2_mod, pairwise ~ matrix, param = 'beta'))

$log10n0
$emmeans
  matrix emmean    SE  df lower.CL upper.CL
ps      7.29 0.182 113    6.93    7.65
s       7.04 0.184 113    6.68    7.41

Confidence level used: 0.95

$contrasts
  contrast estimate    SE  df t.ratio p.value
ps - s      0.245 0.259 113    0.946  0.3460

$TD
$emmeans

```

	matrix	emmean	SE	df	lower.CL	upper.CL
ps		2.13	0.567	113	1.005	3.25
s		2.18	0.617	113	0.956	3.40

Confidence level used: 0.95

\$contrasts

	contrast	estimate	SE	df	t.ratio	p.value
ps - s		-0.0515	0.838	113	-0.061	0.9510

\$beta

\$emmeans

	matrix	emmean	SE	df	lower.CL	upper.CL
ps		0.657	0.0617	113	0.534	0.779
s		0.640	0.0641	113	0.513	0.767

Confidence level used: 0.95

\$contrasts

	contrast	estimate	SE	df	t.ratio	p.value
ps - s		0.0166	0.089	113	0.187	0.8520

Overall marginal means

emmeans (m2_mod, ~ 1, param = 'log10n0')

	1	emmean	SE	df	lower.CL	upper.CL
overall		7.16	0.129	113	6.91	7.42

Results are averaged over the levels of: matrix

Confidence level used: 0.95

emmeans (m2_mod, ~ 1, param = 'beta')

	1	emmean	SE	df	lower.CL	upper.CL
overall		0.648	0.0445	113	0.56	0.736

Results are averaged over the levels of: matrix

Confidence level used: 0.95

(TD_values <- emmeans (m2_mod, ~ 1, param = 'TD'))

	1	emmean	SE	df	lower.CL	upper.CL
overall		2.15	0.419	113	1.32	2.98

Results are averaged over the levels of: matrix

Confidence level used: 0.95

saveRDS (TD_values, here('obj2', 'results', "TD_values.RDS"))

3.2.3 Assess covariance/collinearity

```
set.seed(SEED)
iter = ITER

## Variance-covariance matrix
m2_mod <- readRDS(here('obj2', 'results', 'ecoli_mod_sum.RDS'))
m2_mod_coef <- coef (m2_mod)

(mod_vcov_joint <- vcov(m2_mod))

      log10n0.(Intercept) log10n0.matrixs TD.(Intercept)
log10n0.(Intercept)      0.03293      -0.03293      -0.0901
log10n0.matrixs         -0.03293       0.06693       0.0901
TD.(Intercept)          -0.09011       0.09011       0.3212
TD.matrixs              0.09011      -0.19030      -0.3212
beta.(Intercept)        -0.00887       0.00887       0.0343
beta.matrixs            0.00887      -0.01827      -0.0343

      TD.matrixs beta.(Intercept) beta.matrixs
log10n0.(Intercept)  0.0901      -0.00887    0.00887
log10n0.matrixs     -0.1903       0.00887   -0.01827
TD.(Intercept)      -0.3212       0.03432   -0.03432
TD.matrixs          0.7021      -0.03432    0.07310
beta.(Intercept)    -0.0343       0.00381   -0.00381
beta.matrixs        0.0731      -0.00381    0.00792

mod_vcov_indep <- diag(diag(mod_vcov_joint))
dimnames(mod_vcov_indep) <- dimnames(mod_vcov_joint)

## Draw of coefficients
draw_joint <- MASS::mvrnorm(n=iter, mu = m2_mod_coef, Sigma = mod_vcov_joint)
|>
  as.data.frame()

## Assuming no covariance
draw_indep <- MASS::mvrnorm(n=iter, mu = m2_mod_coef, Sigma = mod_vcov_indep)
|>
  as.data.frame()

## Extract TD values
TD_joint <- cbind (
  ps_joint = draw_joint[, "TD.(Intercept)"],
  s_joint = draw_joint[, "TD.(Intercept)"] + draw_joint[, "TD.matrixs"]
)

colMeans(TD_joint <= 0)

ps_joint  s_joint
2e-04     0e+00
```

```

TD_indep <- cbind (
  ps_indep = draw_indep[, "TD.(Intercept)"],
  s_indep = draw_indep[, "TD.(Intercept)"] + draw_indep[, "TD.matrixs"]
)

colMeans(TD_indep <= 0)

ps_indep  s_indep
0.0000    0.0154

## Calculate mean and 95% QIs
QIs <- function(x) c(
  mean=mean(x),
  lwr=quantile(x, 0.025),
  upr=quantile(x, 0.975))

round(t(apply(TD_joint, 2, QIs)),2)

      mean lwr.2.5% upr.97.5%
ps_joint 2.12      1.01      3.23
s_joint  2.18      0.95      3.37

round(t(apply(TD_indep, 2, QIs)),2)

      mean lwr.2.5% upr.97.5%
ps_indep 2.14      1.06      3.22
s_indep  2.19      0.25      4.14

## Agin compare with emmeans
emmeans (m2_mod, pairwise ~ matrix, param = 'TD')$emmeans

matrix emmean    SE  df lower.CL upper.CL
ps      2.13 0.567 113    1.005    3.25
s       2.18 0.617 113    0.956    3.40

Confidence level used: 0.95

saveRDS(mod_vcov_joint, here('obj2', 'results', 'ecoli_sim_mvrnorm.RDS'))

# Remove selected objects from the environment
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
  envir = .GlobalEnv)

3.2.4 Plot microbial inactivation
m2 <- readRDS(here('obj2', 'data', 'obj2_dat.RDS')) |>
  dplyr::filter(microbe == 'ecoli') |>
  dplyr::mutate(matrix = factor(matrix,
                                labels = c('Soil with plant', 'Soil'))) |>
  droplevels()

m2_preds <- readRDS(here('Obj2', 'results', 'ecoli_mod_preds.RDS'))

```



```

m2_preds$matrix <- factor(m2_preds$matrix, levels = c("ps", "s"),
                          labels = c("Soil with plant", "Soil"))

ec_m2 <- ggplot(m2_preds, aes(x = week, color = matrix, fill = matrix,
linetype = matrix)) +
  geom_hline(aes(yintercept = LOD$lod[LOD$microbe == "ecoli_plate"]),
            color = 'black', linetype = 3) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high),
            alpha = 0.2,
            color = NA) +
  geom_line(aes(y = estimate), linewidth = 0.7) +
  geom_point(data = m2,
            aes(y = survival, shape = matrix),
            size = 1,
            position = position_jitter(width = 0.03)) +
  guides(
    color = guide_legend(title = 'Matrix:'),
    linetype = guide_legend(title = 'Matrix:'),
    shape = guide_legend(title = 'Matrix:')
  ) +
  labs(x = "",
       y = bquote(Log[10]~CFU~per~g)) +
  scale_fill_manual(values = c("#00529b", "#fb6502")) +
  scale_color_manual(values = c("#00529b", "#fb6502")) +
  scale_shape_manual(values = c(1, 2)) +
  scale_linetype_manual(values = c("longdash", "dotdash")) +
  scale_x_continuous(breaks = seq(0, 30, 3), labels = seq(0, 30, 3)) +
  scale_y_continuous(breaks = seq(0, 9, 1), labels = seq(0, 9, 1), limits =
c(0.5, 9)) +
  theme(text = element_text(family = "Arial"),
        legend.position="none",
        plot.title = element_text(color = "black", size = 7, face = "bold",
hjust = 0.5),
        axis.text.x = element_text(size = 7, color = "black", face = "bold",
hjust = 0.5, vjust = 0.5),
        axis.text.y = element_text(size = 7, color = "black", face = "bold"),
        axis.title.x = element_text(size = 7, vjust = -1),
        axis.title.y = element_text(size = 7, vjust = 2),
        panel.border = element_rect(color = "black", fill = NA, linewidth =
1),
        panel.grid.major = element_line(color = "gray", linewidth = 0.1,
linetype = "longdash"),
        panel.grid.minor = element_line(color = "lightgray", linewidth = 0.1,
linetype = "dotted"),
        panel.background = element_rect(fill = "transparent"),
        legend.background = element_rect(fill = "transparent", color =
"transparent"),
        plot.background = element_rect(fill = "transparent", color =
"transparent"))

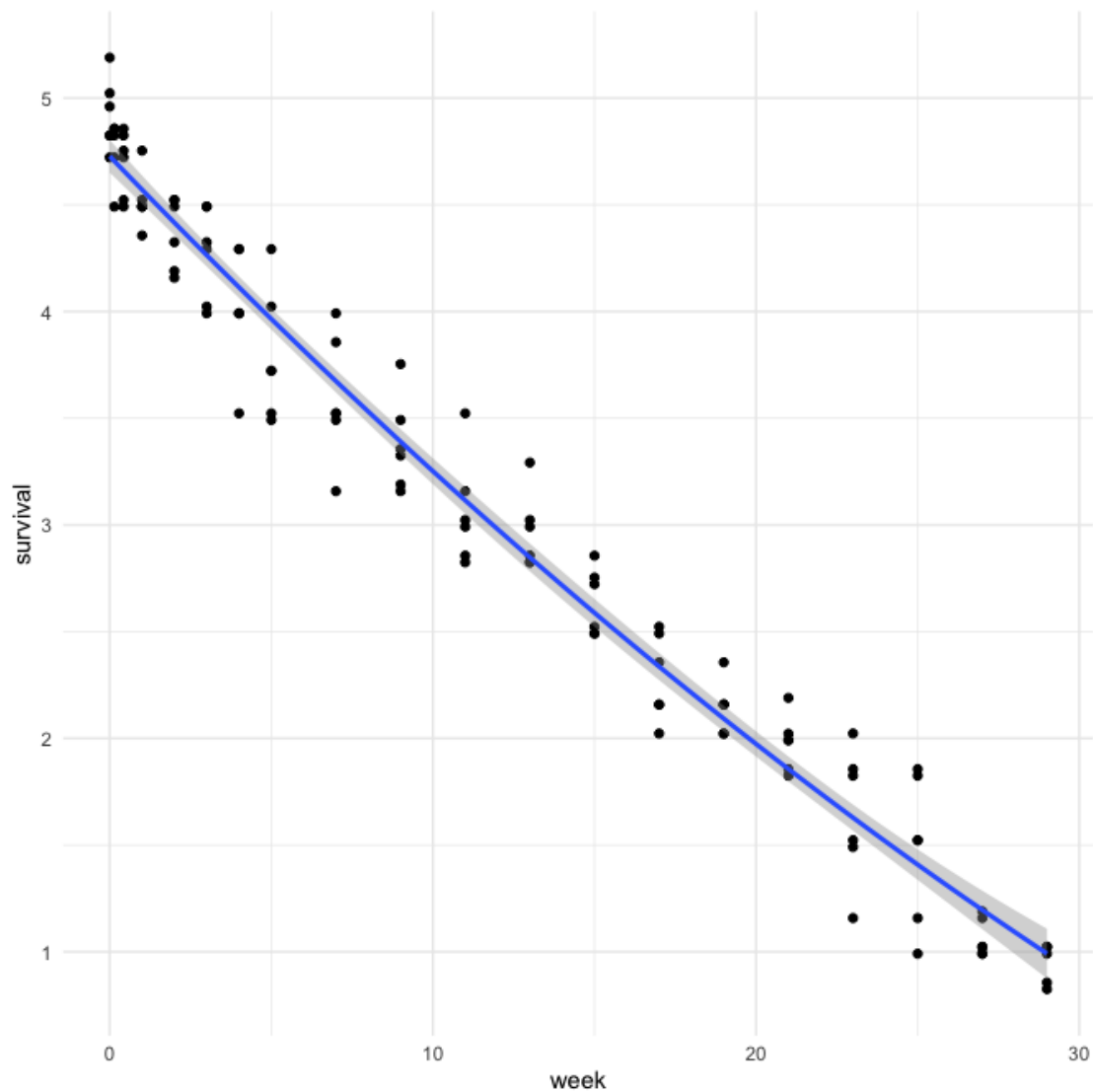
```

3.3 Tulane virus TCID50

3.3.1 Fitting models

```
m2 <- readRDS(here('obj2', 'data', 'Obj2_dat.RDS')) |>
  dplyr::filter(microbe=='tv', assay=='tcid50') |>
  droplevels()

# Initial visual inspection
ggplot(m2, aes(x = week, y = survival)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
  theme_minimal()
```



```
# Levels
contrasts(m2$matrix)
```

```

s
ps 0
s 1

m2_mod_null = nlme::gnls(survival ~ ll(x = week,
                                     log10n0 = log10n0,
                                     alpha = alpha),
                        data = m2,
                        params = log10n0+alpha ~ 1,
                        start = list(log10n0 = 5,
                                     alpha = 0.2))

m2_mod_ll = nlme::gnls(survival ~ ll(x = week,
                                     log10n0 = log10n0,
                                     alpha = alpha),
                      data = m2,
                      params = log10n0+alpha ~ matrix,
                      start = list(log10n0 = c(5, 0),
                                    alpha = c(0.2, 0)))

m2_mod_llrepar = nlme::gnls(survival ~ llrepar(x = week,
                                              log10n0 = log10n0,
                                              TD = TD),
                           data = m2,
                           params = log10n0+TD ~ matrix,
                           start = list(log10n0 = c(5, 0),
                                          TD = c(7, 0)))

m2_mod_wb = nlme::gnls(survival ~ wb(x = week,
                                     log10n0 = log10n0,
                                     alpha = alpha,
                                     beta = beta),
                      data = m2,
                      params = log10n0+alpha+beta ~ matrix,
                      start = list(log10n0 = c(5, 0),
                                    alpha = c(0.2, 0),
                                    beta = c(0.2, 0)))

m2_mod_wbrepar = nlme::gnls(survival ~ wbrepar(x = week,
                                              log10n0 = log10n0,
                                              TD = TD,
                                              beta = beta),
                           data = m2,
                           params = log10n0+TD+beta ~ matrix,
                           start = list(log10n0 = c(5, 0),
                                          TD = c(5, 0),
                                          beta = c(2, 0)))

m2_mod_lg = nlme::gnls(survival ~ lg(x = week,
                                     log10n0 = log10n0,

```

```

        kappa = kappa,
        sigma = sigma),
  data = m2,
  params = log10n0+kappa+sigma ~ matrix,
  start = list(log10n0 = c(5, 0),
               kappa = c(1, 0),
               sigma = c(1, 0)))

m2_mod_lgpar = nlme::gnls(survival ~ lgpar(x = week,
        log10n0 = log10n0,
        sigma = sigma,
        TD = TD),
  data = m2,
  params = log10n0+sigma+TD ~ matrix,
  start = list(log10n0 = c(5, 0),
               sigma = c(1, 0),
               TD = c(6, 0)))

```

3.3.2 Model performance and inference

```

# Model comparisons
performance::compare_performance (
  m2_mod_null, m2_mod_ll, m2_mod_llrepar,
  m2_mod_lg, m2_mod_lgpar, m2_mod_wb, m2_mod_wbrepar,
  metrics = c('AICc'))

```

Comparison of Model Performance Indices

Name	Model	AIC (weights)	AICc (weights)	BIC (weights)
R2				

--				
m2_mod_null	gnls	-11.0 (<.001)	-10.8 (<.001)	-2.6 (0.001)
0.969				
m2_mod_ll	gnls	-15.6 (<.001)	-15.1 (<.001)	-1.7 (<.001)
0.971				
m2_mod_llrepar	gnls	-15.6 (<.001)	-15.1 (<.001)	-1.7 (<.001)
0.971				
m2_mod_lg	gnls	51.2 (<.001)	52.2 (<.001)	70.7 (<.001)
0.951				
m2_mod_lgpar	gnls	51.2 (<.001)	52.2 (<.001)	70.7 (<.001)
0.951				
m2_mod_wb	gnls	-34.5 (0.500)	-33.5 (0.500)	-15.0 (0.499)
0.976				
m2_mod_wbrepar	gnls	-34.5 (0.500)	-33.5 (0.500)	-15.0 (0.499)
0.976				

Name	RMSE	Sigma

m2_mod_null	0.225	0.227
m2_mod_ll	0.217	0.221

```

m2_mod_llrepar | 0.217 | 0.221
m2_mod_lg      | 0.283 | 0.290
m2_mod_lgpar   | 0.283 | 0.290
m2_mod_wb      | 0.198 | 0.203
m2_mod_wbrepar | 0.198 | 0.203

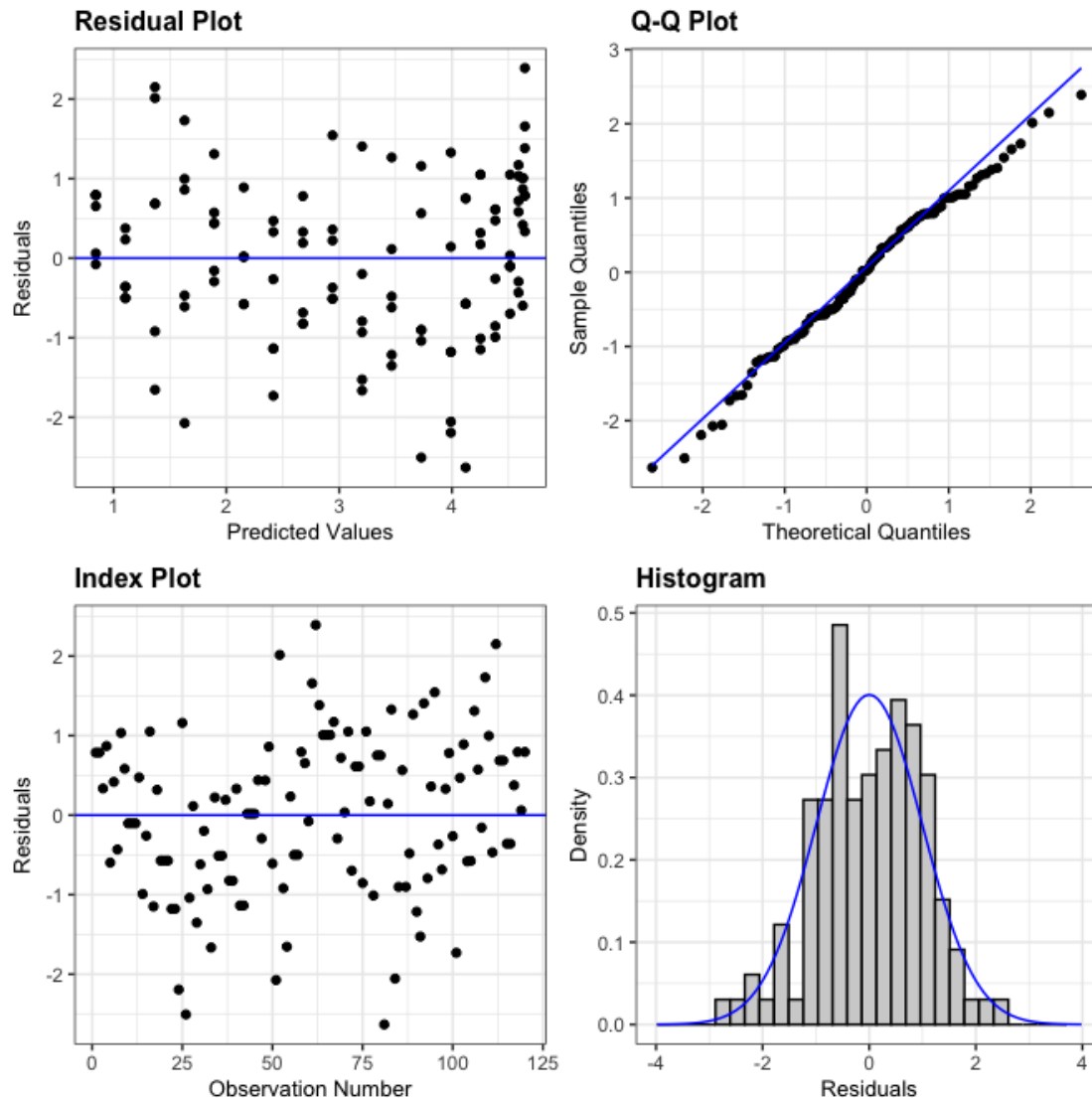
```

```
# Residual plots
```

```

m2_res_null = resid(m2_mod_null, type = 'normalized')
m2_fit_null = fitted(m2_mod_null)
resid_auxpanel(m2_res_null, m2_fit_null)

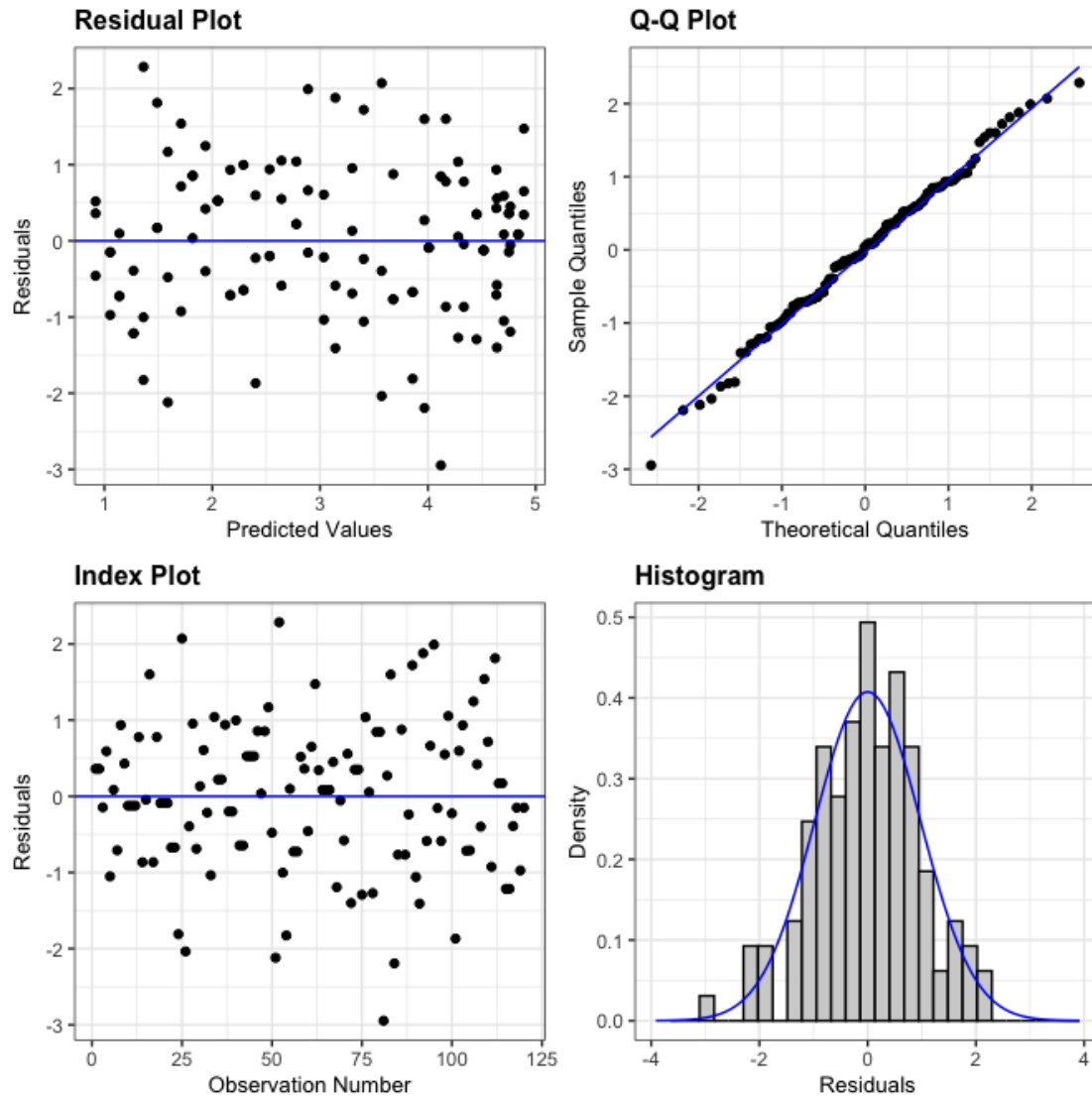
```



```

m2_res_wbrepar = resid(m2_mod_wbrepar, type = 'normalized')
m2_fit_wbrepar = fitted(m2_mod_wbrepar)
resid_auxpanel(m2_res_wbrepar, m2_fit_wbrepar)

```



```
pres_mod <- m2_mod_wbrepair
```

```
# gnls model tuning
m2_mod_exp = update(pres_mod, weights = varExp())
m2_mod_exp_wk = update(pres_mod, weights = varExp(form = ~week))
m2_mod_idt_mtx = update(pres_mod, weights = varIdent(form = ~1 | matrix))
#m2_mod_idt_mic = update(pres_mod, weights = varIdent(form = ~1 | microbe))
m2_mod_pow = update(pres_mod, weights = varPower())

#m2_mod_combined1 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
microbe), varExp()))
#m2_mod_combined2 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
microbe), varExp(form = ~ week)))
m2_mod_combined3 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
matrix), varExp()))
m2_mod_combined4 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
```

```
matrix), varExp(form = ~ week)))
```

```
compare_performance(
  pres_mod,
  m2_mod_exp,
  m2_mod_exp_wk,
  m2_mod_idt_mtx,
  #m2_mod_idt_mic,
  m2_mod_pow,
  #m2_mod_combined1,
  #m2_mod_combined2,
  m2_mod_combined3,
  m2_mod_combined4,
  metrics = c('AICc'))
```

```
# Comparison of Model Performance Indices
```

Name	Model	AIC (weights)	AICc (weights)	BIC (weights)
pres_mod	gnls	-34.5 (0.264)	-33.5 (0.304)	-15.0 (0.647)
m2_mod_exp	gnls	-32.8 (0.113)	-31.5 (0.112)	-10.5 (0.069)
m2_mod_exp_wk	gnls	-32.6 (0.107)	-31.4 (0.106)	-10.3 (0.065)
m2_mod_idt_mtx	gnls	-34.0 (0.211)	-32.7 (0.209)	-11.7 (0.128)
m2_mod_pow	gnls	-32.5 (0.100)	-31.2 (0.099)	-10.2 (0.061)
m2_mod_combined3	gnls	-32.7 (0.109)	-31.1 (0.091)	-7.6 (0.016)
m2_mod_combined4	gnls	-32.4 (0.095)	-30.8 (0.080)	-7.3 (0.014)

Name	R2	RMSE	Sigma
pres_mod	0.976	0.198	0.203
m2_mod_exp	0.976	0.198	0.224
m2_mod_exp_wk	0.976	0.198	0.195
m2_mod_idt_mtx	0.976	0.198	0.186
m2_mod_pow	0.976	0.198	0.211
m2_mod_combined3	0.976	0.198	0.214
m2_mod_combined4	0.976	0.198	0.174

```
(m2_mod <- pres_mod)
```

```
Generalized nonlinear least squares fit
```

```
Model: survival ~ wbrepar(x = week, log10n0 = log10n0, TD = TD, beta =
beta)
```

```
Data: m2
```

```
Log-likelihood: 24.2
```

```
Coefficients:
```

```
log10n0.(Intercept)      log10n0.matrixs      TD.(Intercept)
TD.matrixs
4.7517                  0.1390                  5.7305      -
0.2105
```

beta.(Intercept)	beta.matrixs
0.8288	-0.0182

Degrees of freedom: 120 total; 114 residual
Residual standard error: 0.203

`summary (m2_mod)`

Generalized nonlinear least squares fit

Model: survival ~ wbrepar(x = week, log10n0 = log10n0, TD = TD, beta = beta)

Data: m2

AIC	BIC	logLik
-34.5	-15	24.2

Coefficients:

	Value	Std.Error	t-value	p-value
log10n0.(Intercept)	4.75	0.067	70.5	0.000
log10n0.matrixs	0.14	0.096	1.4	0.150
TD.(Intercept)	5.73	0.560	10.2	0.000
TD.matrixs	-0.21	0.788	-0.3	0.790
beta.(Intercept)	0.83	0.046	17.8	0.000
beta.matrixs	-0.02	0.065	-0.3	0.781

Correlation:

	l100.(lg100.	TD.(I)	TD.mtr	bt.(I)
log10n0.matrixs	-0.702				
TD.(Intercept)	-0.869	0.610			
TD.matrixs	0.617	-0.870	-0.710		
beta.(Intercept)	-0.761	0.534	0.966	-0.686	
beta.matrixs	0.543	-0.763	-0.690	0.967	-0.714

Standardized residuals:

Min	Q1	Med	Q3	Max
-2.9461	-0.6764	0.0461	0.6183	2.2847

Residual standard error: 0.203

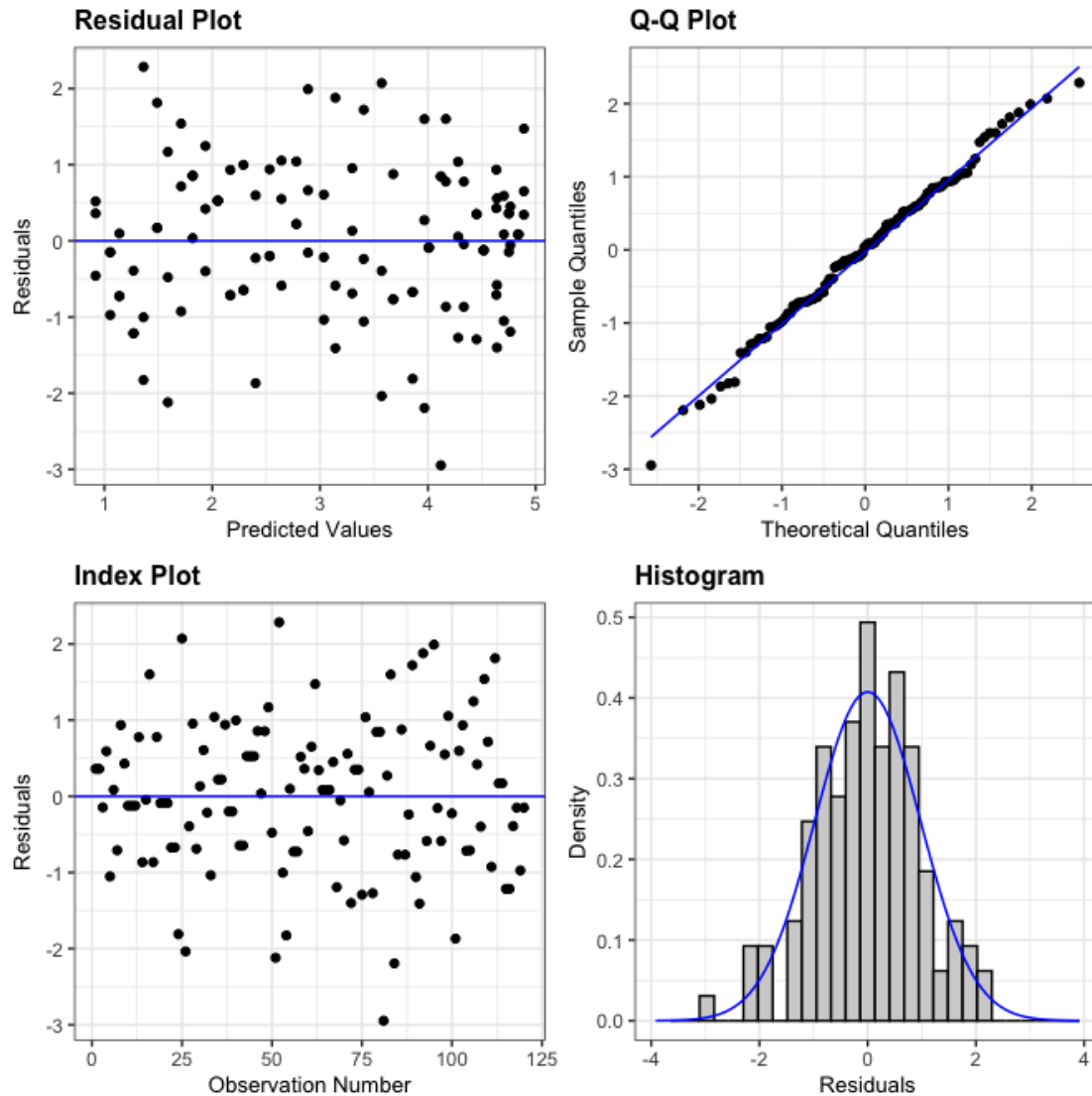
Degrees of freedom: 120 total; 114 residual

Residual plots

`m2_res_m2_mod = resid(m2_mod, type = 'normalized')`

`m2_fit_m2_mod = fitted(m2_mod)`

`resid_auxpanel(m2_res_m2_mod, m2_fit_m2_mod)`



```
saveRDS(m2_mod, here('obj2', 'results', 'tv_tcid50_mod_sum.RDS'))

# Marginal means (prediction)
m2_mod <- readRDS(here('obj2', 'results', 'tv_tcid50_mod_sum.RDS'))

expand.grid(
  matrix = levels(m2$matrix),
  microbe = levels(m2$microbe),
  assay = levels(m2$assay),
  week = seq(0, max(m2$week), 0.5)) |>
marginaleffects::avg_predictions(
  model=m2_mod,
  conf_level = 0.95,
  by = c('matrix', 'week')) |>
saveRDS(here('obj2', 'results', 'tv_tcid50_mod_preds.RDS'))
```

```

# Extract parameters (coefficients) and their 95% CI
m2_mod_coef = coef(m2_mod)
m2_mod_confint = confint(m2_mod)

as.data.frame (cbind(
  Param = names (m2_mod_coef),
  Est = round (m2_mod_coef, 3),
  lwr = round (m2_mod_confint[,1], 3),
  upr = round (m2_mod_confint[,2], 3)
))

```

	Param	Est	lwr	upr
log10n0.(Intercept)	log10n0.(Intercept)	4.752	4.62	4.884
log10n0.matrixs	log10n0.matrixs	0.139	-0.049	0.327
TD.(Intercept)	TD.(Intercept)	5.731	4.633	6.828
TD.matrixs	TD.matrixs	-0.21	-1.756	1.335
beta.(Intercept)	beta.(Intercept)	0.829	0.738	0.92
beta.matrixs	beta.matrixs	-0.018	-0.146	0.109

```

# Joint tests
list(
  log10n0 = joint_tests (m2_mod, param = 'log10n0'),
  TD = joint_tests (m2_mod, param = 'TD'),
  beta = joint_tests (m2_mod, param = 'beta'))

```

```

$log10n0
model term df1 df2 F.ratio p.value
matrix      1 113   2.096  0.1504

```

```

$TD
model term df1 df2 F.ratio p.value
matrix      1 113   0.071  0.7900

```

```

$beta
model term df1 df2 F.ratio p.value
matrix      1 113   0.078  0.7810

```

```

# Estimated marginal (group level) means and contrasts
list(
  log10n0 = emmeans (m2_mod, pairwise ~ matrix, param = 'log10n0'),
  TD = emmeans (m2_mod, pairwise ~ matrix, param = 'TD'),
  beta = emmeans (m2_mod, pairwise ~ matrix, param = 'beta'))

```

```

$log10n0
$emmeans
matrix emmean      SE  df lower.CL upper.CL
ps      4.75 0.0674 113     4.62     4.89
s       4.89 0.0684 113     4.76     5.03

```

Confidence level used: 0.95

\$contrasts

contrast	estimate	SE	df	t.ratio	p.value
ps - s	-0.139	0.096	113	-1.448	0.1504

\$TD

\$emmeans

matrix	emmean	SE	df	lower.CL	upper.CL
ps	5.73	0.560	113	4.62	6.84
s	5.52	0.555	113	4.42	6.62

Confidence level used: 0.95

\$contrasts

contrast	estimate	SE	df	t.ratio	p.value
ps - s	0.21	0.788	113	0.267	0.7900

\$beta

\$emmeans

matrix	emmean	SE	df	lower.CL	upper.CL
ps	0.829	0.0465	113	0.737	0.921
s	0.811	0.0456	113	0.720	0.901

Confidence level used: 0.95

\$contrasts

contrast	estimate	SE	df	t.ratio	p.value
ps - s	0.0182	0.0651	113	0.279	0.7810

Overall marginal means

emmeans (m2_mod, ~ 1, param = 'log10n0')

1	emmean	SE	df	lower.CL	upper.CL
overall	4.82	0.048	113	4.73	4.92

Results are averaged over the levels of: matrix

Confidence level used: 0.95

emmeans (m2_mod, ~ 1, param = 'TD')

1	emmean	SE	df	lower.CL	upper.CL
overall	5.63	0.394	113	4.84	6.41

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ 1, param = 'beta')

1      emmean      SE  df lower.CL upper.CL
overall  0.82 0.0326 113    0.755    0.884
```

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
TD_values <- list (
  ecoli_plate_1D = readRDS (here('obj2', 'results', 'TD_values.RDS')),
  tv_tcid50_1D = emmeans (m2_mod, ~ 1, param = 'beta')
)
```

```
saveRDS (TD_values, here('obj2', 'results', "TD_values.RDS"))
```

3.3.3 Assess covariance/collinearity

```
set.seed(SEED)
iter = ITER
```

```
## Variance-covariance matrix
m2_mod <- readRDS (here('obj2', 'results', 'tv_tcid50_mod_sum.RDS'))
m2_mod_coef <- coef (m2_mod)
```

```
(mod_vcov_joint <- vcov(m2_mod))
```

	log10n0.(Intercept)	log10n0.matrixs	TD.(Intercept)
log10n0.(Intercept)	0.00455	-0.00455	-0.0328
log10n0.matrixs	-0.00455	0.00922	0.0328
TD.(Intercept)	-0.03280	0.03280	0.3133
TD.matrixs	0.03280	-0.06589	-0.3133
beta.(Intercept)	-0.00239	0.00239	0.0251
beta.matrixs	0.00239	-0.00477	-0.0251
	TD.matrixs	beta.(Intercept)	beta.matrixs
log10n0.(Intercept)	0.0328	-0.00239	0.00239
log10n0.matrixs	-0.0659	0.00239	-0.00477
TD.(Intercept)	-0.3133	0.02515	-0.02515
TD.matrixs	0.6217	-0.02515	0.04964
beta.(Intercept)	-0.0251	0.00216	-0.00216
beta.matrixs	0.0496	-0.00216	0.00424

```
mod_vcov_indep <- diag(diag(mod_vcov_joint))
dimnames(mod_vcov_indep) <- dimnames(mod_vcov_joint)
```

```
## Draw of coefficients
```

```
iter = 5000
draw_joint <- MASS::mvrnorm(n=iter, mu = m2_mod_coef, Sigma = mod_vcov_joint)
|>
  as.data.frame()
```

```

## Assuming no covariance
draw_indep <- MASS::mvrnorm(n=iter, mu = m2_mod_coef, Sigma = mod_vcov_indep)
|>
  as.data.frame()

## Extract TD values
TD_joint <- cbind (
  ps_joint = draw_joint[, "TD.(Intercept)"],
  s_joint = draw_joint[, "TD.(Intercept)"] + draw_joint[, "TD.matrixs"]
)

colMeans(TD_joint <= 0)

ps_joint  s_joint
      0      0

TD_indep <- cbind (
  ps_indep = draw_indep[, "TD.(Intercept)"],
  s_indep = draw_indep[, "TD.(Intercept)"] + draw_indep[, "TD.matrixs"]
)

colMeans(TD_indep <= 0)

ps_indep  s_indep
      0      0

## Calculate mean and 95% QIs
QIs <- function(x) c(
  mean=mean(x),
  lwr=quantile(x, 0.025),
  upr=quantile(x, 0.975))

round(t(apply(TD_joint, 2, QIs)),2)

      mean lwr.2.5% upr.97.5%
ps_joint  5.72     4.62     6.82
s_joint    5.52     4.41     6.58

round(t(apply(TD_indep, 2, QIs)),2)

      mean lwr.2.5% upr.97.5%
ps_indep  5.74     4.68     6.81
s_indep    5.53     3.67     7.39

saveRDS(mod_vcov_joint, here('obj2', 'results', 'tv_tcid50_sim_mvrnorm.RDS'))

## Agin compare with emmeans
emmeans (m2_mod, pairwise ~ matrix, param = 'TD')$emmeans

```

	matrix	emmean	SE	df	lower.CL	upper.CL
ps		5.73	0.560	113	4.62	6.84
s		5.52	0.555	113	4.42	6.62

Confidence level used: 0.95

```
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
    envir = .GlobalEnv)
```

3.3.4 Plot microbial inactivation

```
m2 <- readRDS(here('obj2', 'data', 'Obj2_dat.RDS')) |>
  dplyr::filter(microbe=='tv', assay=='tcid50') |>
  dplyr::mutate(matrix = factor(matrix,
                                labels = c('Soil with plant', 'Soil'))) |>
  droplevels()

m2_preds = readRDS(here('obj2', 'results', 'tv_tcid50_mod_preds.RDS'))

m2_preds$matrix <- factor(m2_preds$matrix, levels = c("ps", "s"),
                          labels = c("Soil with plant", "Soil"))

microb_labs = c(tv = 'Tulane virus')

tv_m2 <- ggplot(m2_preds, aes(x = week, color = matrix, fill = matrix,
linetype = matrix)) +
  geom_hline(aes(yintercept = LOD$lod[LOD$microbe == "tv_tcid50"]),
             color = 'black', linetype = 3) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high,
                 group = matrix),
             alpha = 0.2,
             color = NA) +
  geom_line(aes(y = estimate), linewidth = 0.7) +
  geom_point(data = m2,
             aes(y = survival, shape = matrix),
             size = 1,
             position = position_jitter(width = 0.03)) +
  guides(
    fill = guide_legend(title = 'Matrix:'),
    color = guide_legend(title = 'Matrix:'),
    linetype = guide_legend(title = 'Matrix:'),
    shape = guide_legend(title = 'Matrix:')) +
  labs(x = "Time (weeks)",
       y = bquote(Log[10]~TCID[50]~per~g)) +
  scale_fill_manual(values = c("#00529b", "#fb6502")) +
  scale_color_manual(values = c("#00529b", "#fb6502")) +
  scale_shape_manual(values = c(1, 2)) +
  scale_linetype_manual(values = c("longdash", "dotdash")) +
  scale_x_continuous(breaks = seq(0, 30, 3), labels = seq(0, 30, 3)) +
  scale_y_continuous(breaks = seq(0, 6, 1), labels = seq(0, 6, 1), limits =
c(0.5, 6)) +
```

```

theme(text = element_text(family = "Arial"),
      legend.position = 'inside', legend.position.inside = c(0.25, 0.25),
      legend.direction = "vertical",
      legend.text = element_text(size = 5),
      legend.title = element_text(size = 5, face = "bold"),
      legend.key.height = unit(0.4, "cm"),
      plot.title = element_text(color = "black", size = 7, face = "bold",
hjust = 0.5),
      axis.text.x = element_text(size = 7, color = "black", face = "bold",
hjust = 0.5, vjust = 0.5),
      panel.border = element_rect(color = "black", fill = NA, linewidth =
1),
      axis.text.y = element_text(size = 7, color = "black", face = "bold"),
      axis.title.x = element_text(size = 7, vjust = -1),
      axis.title.y = element_text(size = 7, vjust = 2),
      panel.grid.major = element_line(color = "gray", linewidth = 0.1,
linetype = "longdash"),
      panel.grid.minor = element_line(color = "lightgray", linewidth = 0.1,
linetype = "dotted"),
      panel.background = element_rect(fill = "transparent"),
      legend.background = element_rect(fill = "transparent", color =
"transparent"),
      plot.background = element_rect(fill = "transparent", color =
"transparent"))

```

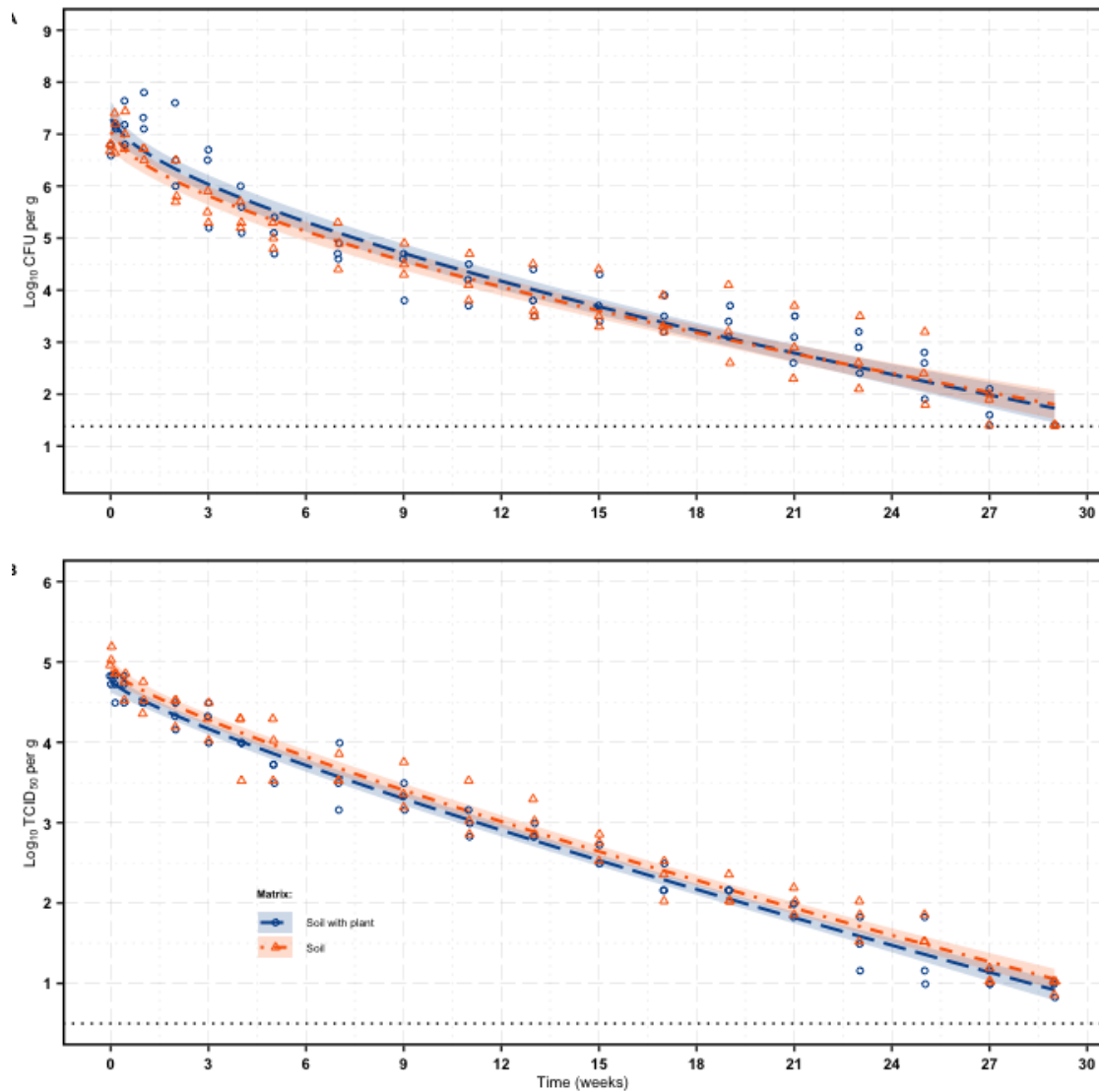
3.4 Combined *E. coli* and Tulane virus plot

```

combined_plots <- ggpubr::ggarrange (ec_m2, tv_m2,
                                     labels = c("A", "B"),
                                     font.label = list(size = 7, face =
"bold"),
                                     label.x = -0.01,
                                     label.y = 0.98,
                                     nrow=2, ncol=1)

combined_plots

```



```
ggsave(
  filename = here('obj2', 'results', 'Fig 1 ec_tv_plots.tiff'),
  plot = combined_plots,
  width = 6,
  height = 9,
  units = "cm",
  dpi = 600
)

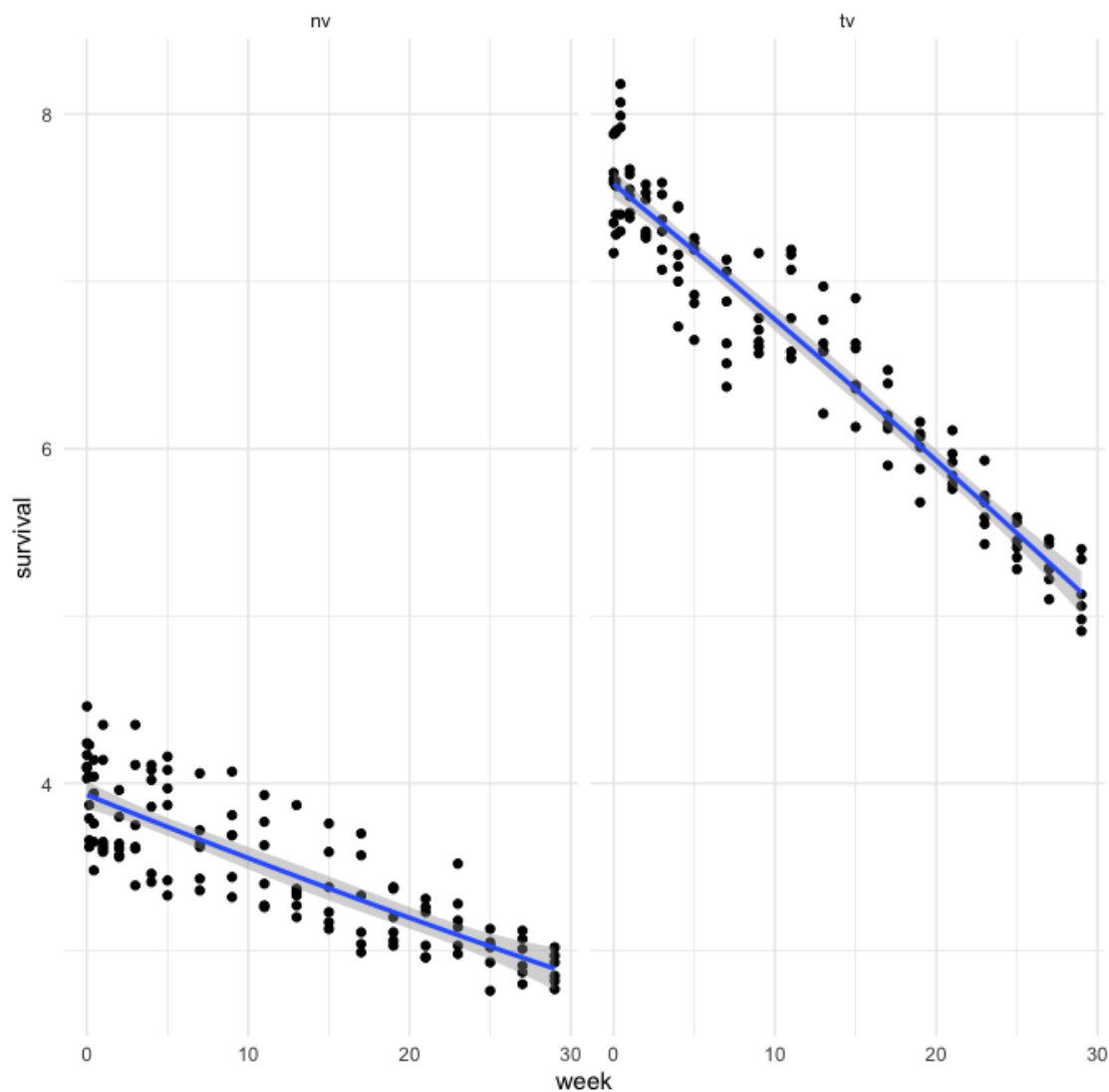
# Remove selected objects from the environment
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
  envir = .GlobalEnv)
```


3.5 TuV and HuNoV - *RNase RT-qPCR*

3.5.1 Fitting models

```
m2 <- readRDS(here('obj2', 'data', 'Obj2_dat.RDS')) |>
  dplyr::filter (assay=='pcr') |>
  droplevels()

# Initial visual inspection
ggplot(m2, aes(x = week, y = survival)) +
  facet_wrap(~ microbe) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ poly(x, 2)) +
  theme_minimal()
```



```
# Levels and combinations
contrasts(m2$matrix)
```

```

      s
ps 0
s 1

contrasts(m2$microbe)

      tv
nv 0
tv 1

levels(interaction(m2$matrix, m2$microbe))

[1] "ps.nv" "s.nv" "ps.tv" "s.tv"

m2_mod_null = nlme::gnls(survival ~ ll(x = week,
                                     log10n0 = log10n0,
                                     alpha = alpha),
                        data = m2,
                        params = log10n0+alpha ~ 1,
                        start = list(log10n0 = 4,
                                     alpha = 0.1))

m2_mod_ll = nlme::gnls(survival ~ ll(x = week,
                                     log10n0 = log10n0,
                                     alpha = alpha),
                      data = m2,
                      params = log10n0+alpha ~ matrix*microbe,
                      start = list(log10n0 = c(4, rep(0,3)),
                                   alpha = c(0.1, rep(0,3))))

m2_mod_llrepar = nlme::gnls(survival ~ llrepar(x = week,
                                              log10n0 = log10n0,
                                              TD = TD),
                          data = m2,
                          params = log10n0+TD ~ matrix*microbe,
                          start = list(log10n0 = c(4, rep(0,3)),
                                       TD = c(25, rep(0,3))))

m2_mod_wb = nlme::gnls(survival ~ wb(x = week,
                                     log10n0 = log10n0,
                                     alpha = alpha,
                                     beta = beta),
                      data = m2,
                      params = log10n0+alpha+beta ~ matrix*microbe,
                      start = list(log10n0 = c(4, rep(0,3)),
                                   alpha = c(4, rep(0,3)),
                                   beta = c(1, rep(0,3))))

m2_mod_wbrepar = nlme::gnls(survival ~ wbrepar(x = week,
                                              log10n0 = log10n0,
                                              TD = TD,

```

```

        beta = beta),
  data = m2,
  params = log10n0+TD+beta ~ matrix*microbe,
  start = list(log10n0 = c(4, rep(0,3)),
               TD = c(1, rep(0,3)),
               beta = c(1, rep(0,3))))

m2_mod_lg = nlme::gnls(survival ~ lg(x = week,
                                   log10n0 = log10n0,
                                   kappa = kappa,
                                   sigma = sigma),
  data = m2,
  params = log10n0+kappa+sigma ~ matrix*microbe,
  start = list(log10n0 = c(4,rep(0,3)),
               kappa = c(2, rep(0,3)),
               sigma = c(1, rep(0,3))))

m2_mod_lgpar = nlme::gnls(survival ~ lgrepar(x = week,
                                             log10n0 = log10n0,
                                             sigma = sigma,
                                             TD = TD),
  data = m2,
  params = log10n0+sigma+TD ~ matrix*microbe,
  start = list(log10n0 = c(4, rep(0,3)),
               sigma = c(1, rep(0,3)),
               TD = c(10, rep(0,3))))

```

3.5.2 Model performance and inferences

Model comparisons

```

performance::compare_performance (
  m2_mod_null, m2_mod_ll, m2_mod_llrepar,
  m2_mod_lg, m2_mod_lgpar, m2_mod_wb, m2_mod_wbrepar,
  metrics = c('AICc'))

```

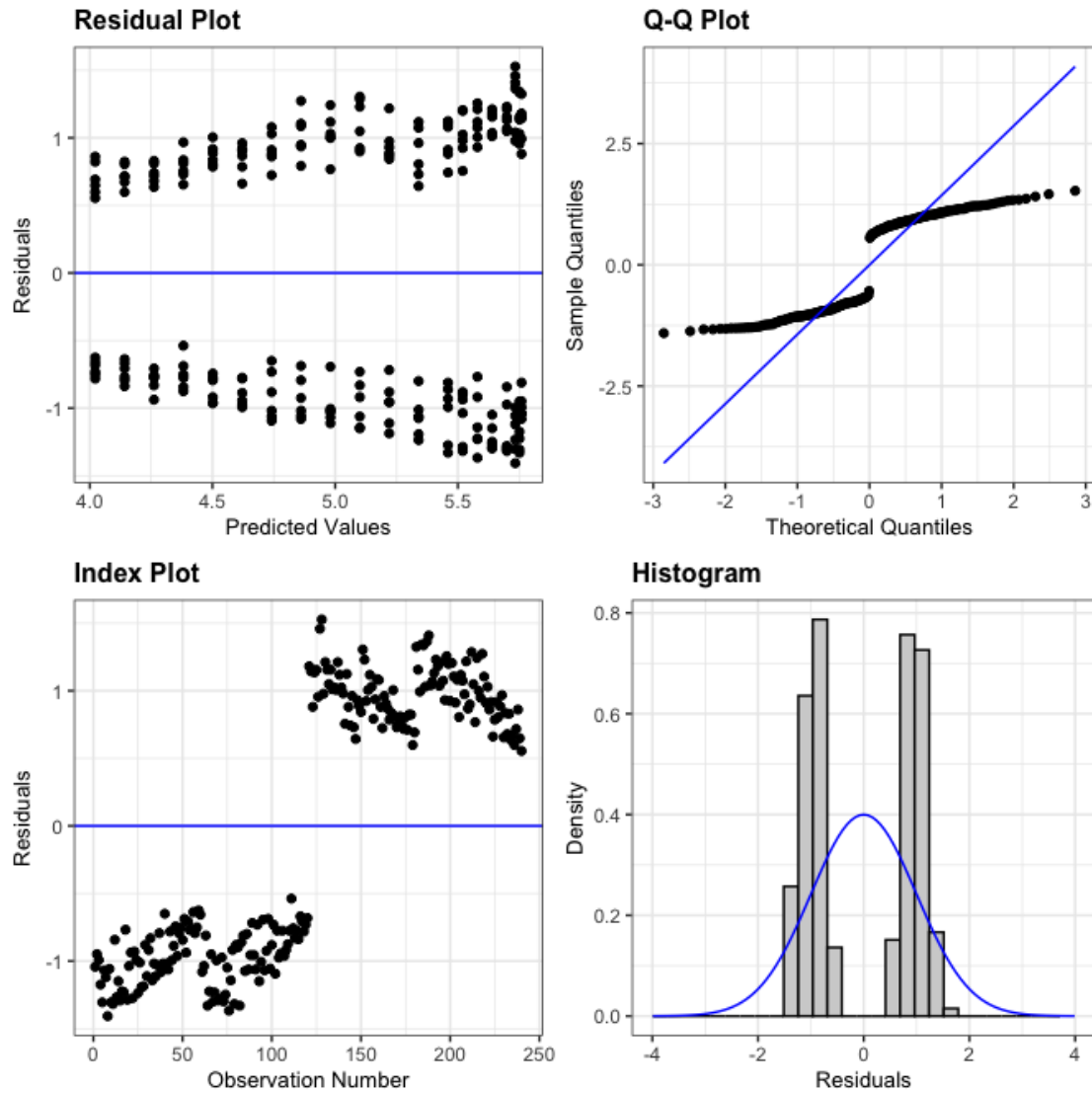
Comparison of Model Performance Indices

Name	Model	AIC (weights)	AICc (weights)	BIC (weights)
m2_mod_null	gnls	911.1 (<.001)	911.2 (<.001)	921.6 (<.001)
m2_mod_ll	gnls	-0.3 (0.461)	0.4 (0.474)	31.0 (0.500)
m2_mod_llrepar	gnls	-0.3 (0.461)	0.4 (0.474)	31.0 (0.500)
m2_mod_lg	gnls	46.3 (<.001)	47.9 (<.001)	91.5 (<.001)
m2_mod_lgpar	gnls	46.3 (<.001)	47.9 (<.001)	91.5 (<.001)

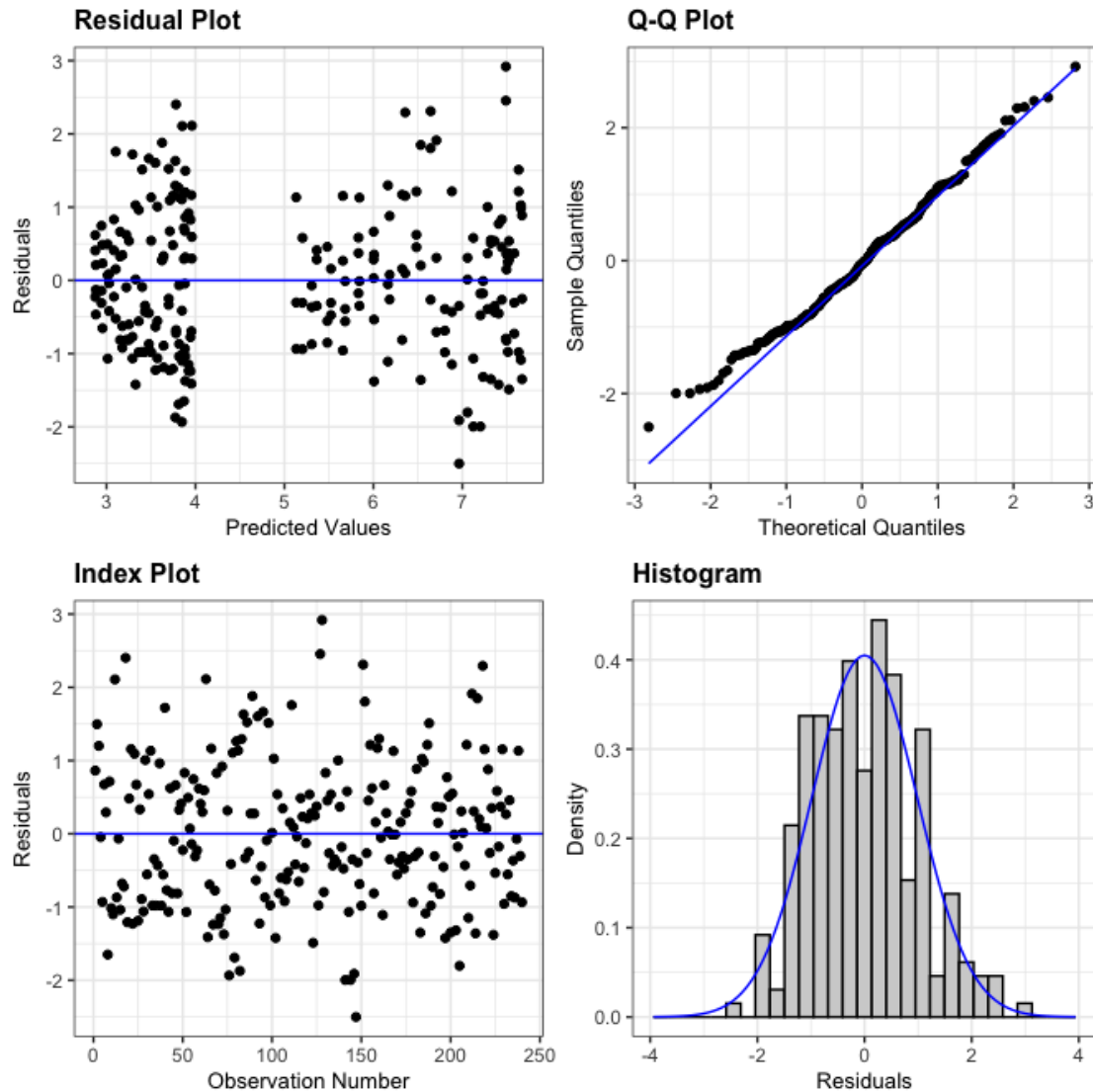
m2_mod_wb	gnls	4.6 (0.039)	6.2 (0.026)	49.9 (<.001)	
0.981					
m2_mod_wbrepair	gnls	4.6 (0.039)	6.2 (0.026)	49.9 (<.001)	
0.981					

Name	RMSE	Sigma
-----	-----	-----
m2_mod_null	1.595	1.601
m2_mod_ll	0.233	0.237
m2_mod_llrepair	0.233	0.237
m2_mod_lg	0.252	0.259
m2_mod_lgpar	0.252	0.259
m2_mod_wb	0.231	0.237
m2_mod_wbrepair	0.231	0.237

```
# Residual plots
m2_res_null = resid(m2_mod_null, type = 'normalized')
m2_fit_null = fitted(m2_mod_null)
resid_auxpanel(m2_res_null, m2_fit_null)
```



```
m2_res_llrepar = resid(m2_mod_llrepar, type = 'normalized')
m2_fit_llrepar = fitted(m2_mod_llrepar)
resid_auxpanel(m2_res_llrepar, m2_fit_llrepar)
```



```
(pres_mod <- m2_mod_llrepar)
```

Generalized nonlinear least squares fit

Model: survival ~ llrepar(x = week, log10n0 = log10n0, TD = TD)

Data: m2

Log-likelihood: 9.17

Coefficients:

log10n0.(Intercept)	log10n0.matrixs	log10n0.microbetv
3.8856	0.0740	3.6372
log10n0.matrixs:microbetv	TD.(Intercept)	TD.matrixs
0.0731	28.6548	-1.7829
TD.microbetv	TD.matrixs:microbetv	
-16.1565	0.7094	

Degrees of freedom: 240 total; 232 residual

Residual standard error: 0.237

```
# gnls model tuning
m2_mod_exp = update(pres_mod, weights = varExp())
m2_mod_exp_wk = update(pres_mod, weights = varExp(form = ~week))
m2_mod_idt_mtx = update(pres_mod, weights = varIdent(form = ~1 | matrix))
m2_mod_idt_mic = update(pres_mod, weights = varIdent(form = ~1 | microbe))
m2_mod_pow = update(pres_mod, weights = varPower())

m2_mod_combined1 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
microbe), varExp()))
m2_mod_combined2 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
microbe), varExp(form = ~ week)))
m2_mod_combined3 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
matrix), varExp()))
m2_mod_combined4 = update(pres_mod, weights = varComb(varIdent(form = ~1 |
matrix), varExp(form = ~ week)))

compare_performance(
  pres_mod,
  m2_mod_exp,
  m2_mod_exp_wk,
  m2_mod_idt_mtx,
  m2_mod_idt_mic,
  m2_mod_pow,
  m2_mod_combined1,
  m2_mod_combined2,
  m2_mod_combined3,
  m2_mod_combined4,
  metrics = c('AICc'))
```

Comparison of Model Performance Indices

Name	Model	AIC (weights)	AICc (weights)	BIC (weights)
pres_mod	gnls	-0.3 (<.001)	0.4 (<.001)	31.0 (<.001)
m2_mod_exp	gnls	0.3 (<.001)	1.2 (<.001)	35.1 (<.001)
m2_mod_exp_wk	gnls	-20.8 (0.545)	-19.8 (0.569)	14.0 (0.872)
m2_mod_idt_mtx	gnls	1.7 (<.001)	2.6 (<.001)	36.5 (<.001)
m2_mod_idt_mic	gnls	1.7 (<.001)	2.6 (<.001)	36.5 (<.001)
m2_mod_pow	gnls	5.6e-03 (<.001)	1.0 (<.001)	34.8 (<.001)
m2_mod_combined1	gnls	-12.7 (0.010)	-11.5 (0.009)	25.6 (0.003)
m2_mod_combined2	gnls	-18.8 (0.206)	-17.7 (0.195)	19.5 (0.058)
m2_mod_combined3	gnls	2.2 (<.001)	3.4 (<.001)	40.5 (<.001)
m2_mod_combined4	gnls	-19.1 (0.240)	-18.0 (0.227)	19.2 (0.067)

Name	R2	RMSE	Sigma
pres_mod	0.981	0.233	0.237
m2_mod_exp	0.981	0.233	0.200
m2_mod_exp_wk	0.981	0.233	0.305
m2_mod_idt_mtx	0.981	0.233	0.236

m2_mod_idt_mic	0.981	0.233	0.238
m2_mod_pow	0.981	0.233	0.178
m2_mod_combined1	0.981	0.233	0.070
m2_mod_combined2	0.981	0.233	0.302
m2_mod_combined3	0.981	0.233	0.197
m2_mod_combined4	0.981	0.233	0.298

```
(m2_mod <- m2_mod_exp_wk)
```

Generalized nonlinear least squares fit

Model: survival ~ llrepar(x = week, log10n0 = log10n0, TD = TD)

Data: m2

Log-likelihood: 20.4

Coefficients:

log10n0.(Intercept)	log10n0.matrixs	log10n0.microbetv
3.868	0.100	3.661
log10n0.matrixs:microbetv	TD.(Intercept)	TD.matrixs
0.058	29.728	-3.303
TD.microbetv	TD.matrixs:microbetv	
-17.292	2.135	

Variance function:

Structure: Exponential of variance covariate

Formula: ~week

Parameter estimates:

expon

-0.0259

Degrees of freedom: 240 total; 232 residual

Residual standard error: 0.305

```
summary (m2_mod)
```

Generalized nonlinear least squares fit

Model: survival ~ llrepar(x = week, log10n0 = log10n0, TD = TD)

Data: m2

AIC BIC logLik

-20.8 14 20.4

Variance function:

Structure: Exponential of variance covariate

Formula: ~week

Parameter estimates:

expon

-0.0259

Coefficients:

	Value	Std.Error	t-value	p-value
log10n0.(Intercept)	3.87	0.05	71.0	0.000
log10n0.matrixs	0.10	0.08	1.3	0.194

log10n0.microbetv	3.66	0.08	47.5	0.000
log10n0.matrixs:microbetv	0.06	0.11	0.5	0.595
TD.(Intercept)	29.73	2.53	11.8	0.000
TD.matrixs	-3.30	3.22	-1.0	0.306
TD.microbetv	-17.29	2.56	-6.7	0.000
TD.matrixs:microbetv	2.14	3.27	0.7	0.514

Correlation:

	l100.(lg10n0.mt	lg10n0.mc	l100.:	TD.(I)	TD.mtr
log10n0.matrixs	-0.707					
log10n0.microbetv	-0.707	0.500				
log10n0.matrixs:microbetv	0.500	-0.707	-0.707			
TD.(Intercept)	-0.865	0.611	0.611	-0.432		
TD.matrixs	0.678	-0.859	-0.480	0.607	-0.785	
TD.microbetv	0.852	-0.602	-0.708	0.500	-0.985	0.773
TD.matrixs:microbetv	-0.668	0.845	0.555	-0.704	0.773	-0.985
	TD.mcr					

log10n0.matrixs
log10n0.microbetv
log10n0.matrixs:microbetv
TD.(Intercept)
TD.matrixs
TD.microbetv
TD.matrixs:microbetv -0.784

Standardized residuals:

Min	Q1	Med	Q3	Max
-2.3405	-0.7486	-0.0182	0.6523	2.6306

Residual standard error: 0.305

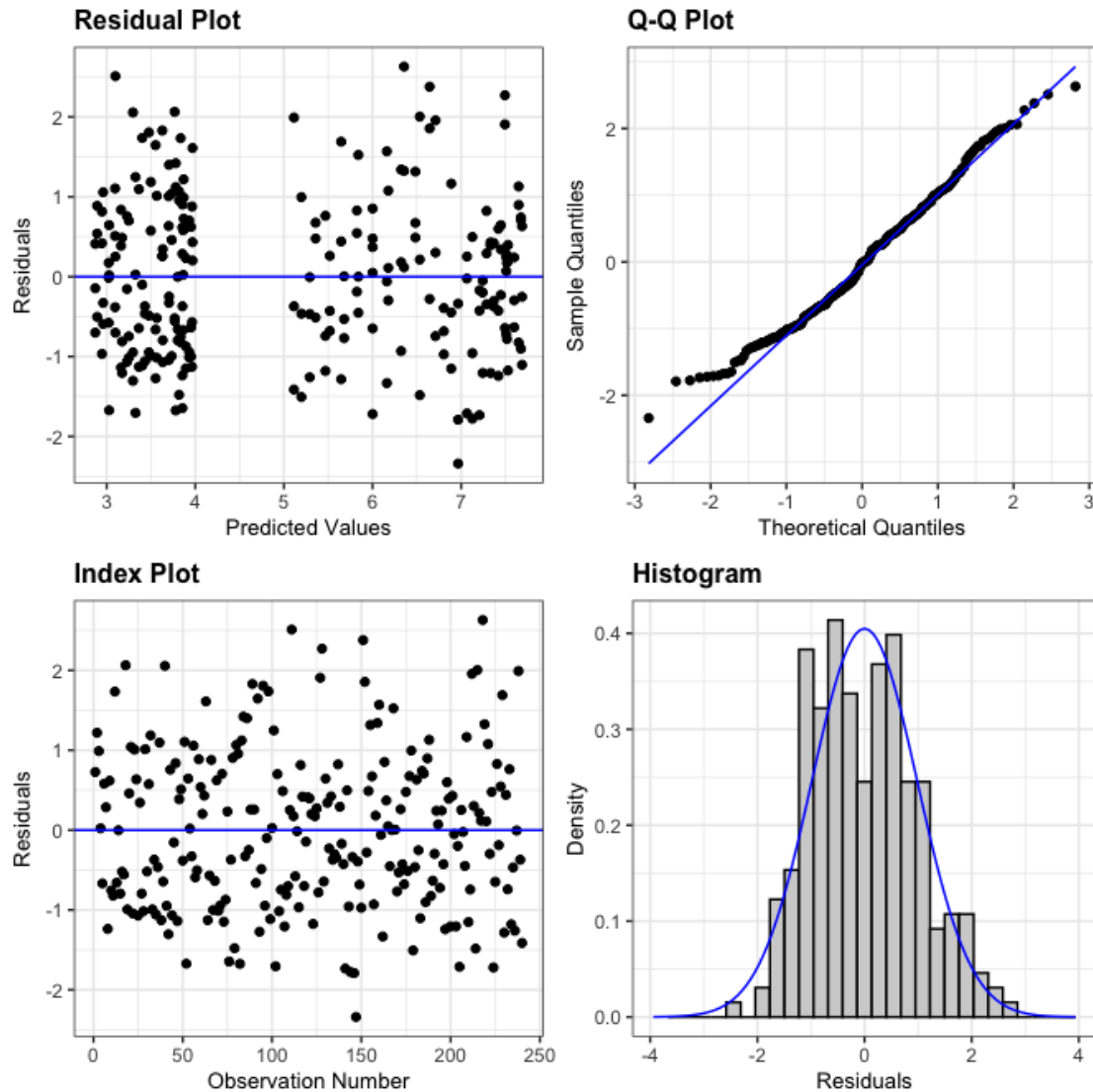
Degrees of freedom: 240 total; 232 residual

Residual plots

m2_res_m2_mod = resid(m2_mod, type = 'normalized')

m2_fit_m2_mod = fitted(m2_mod)

resid_auxpanel(m2_res_m2_mod, m2_fit_m2_mod)



```
saveRDS(m2_mod, here('obj2', 'results', 'nv_tv_pcr_mod_sum.RDS'))

# Marginal means (prediction)
m2_mod <- readRDS(here('obj2', 'results', 'nv_tv_pcr_mod_sum.RDS'))

expand.grid(
  matrix = levels(m2$matrix),
  microbe = levels(m2$microbe),
  assay = levels(m2$assay),
  week = seq(0, max(m2$week), 0.5)) |>
marginaleffects::avg_predictions(
  model=m2_mod,
  conf_level = 0.95,
  by = c('matrix', 'microbe', 'week')) |>
saveRDS(here('obj2', 'results', 'nv_tv_pcr_mod_preds.RDS'))
```

```
# Extract parameters (coefficients) and their 95% CI
m2_mod_coef = coef(m2_mod)
m2_mod_confint = confint(m2_mod)
```

```
as.data.frame (cbind(
  Param = names (m2_mod_coef),
  Est = round (m2_mod_coef, 3),
  lwr = round (m2_mod_confint[,1], 3),
  upr = round (m2_mod_confint[,2], 3)
))
```

	Param	Est	lwr	upr
log10n0.(Intercept)	log10n0.(Intercept)	3.868	3.761	3.975
log10n0.matrixs	log10n0.matrixs	0.1	-0.051	0.251
log10n0.microbetv	log10n0.microbetv	3.661	3.51	3.812
log10n0.matrixs:microbetv	log10n0.matrixs:microbetv	0.058	-0.156	0.272
TD.(Intercept)	TD.(Intercept)	29.728	24.777	34.678
TD.matrixs	TD.matrixs	-3.303	-9.612	3.006
TD.microbetv	TD.microbetv	-17.292	-22.317	-12.266
TD.matrixs:microbetv	TD.matrixs:microbetv	2.135	-4.273	8.543

```
# Estimated marginal (group level) means and contrasts
```

```
## Interaction model
```

```
list(
  log10n0 = emmeans (m2_mod, pairwise ~ matrix*microbe, param = 'log10n0'),
  TD = emmeans (m2_mod, pairwise ~ matrix*microbe, param = 'TD'))
```

```
$log10n0
```

```
$emmeans
```

matrix	microbe	emmean	SE	df	lower.CL	upper.CL
ps	nv	3.87	0.0545	230	3.76	3.98
s	nv	3.97	0.0545	230	3.86	4.08
ps	tv	7.53	0.0545	230	7.42	7.64
s	tv	7.69	0.0545	230	7.58	7.79

```
Confidence level used: 0.95
```

```
$contrasts
```

contrast	estimate	SE	df	t.ratio	p.value
ps nv - s nv	-0.10	0.0771	230	-1.300	0.5630
ps nv - ps tv	-3.66	0.0771	230	-47.500	<.0001
ps nv - s tv	-3.82	0.0771	230	-49.600	<.0001
s nv - ps tv	-3.56	0.0771	230	-46.200	<.0001
s nv - s tv	-3.72	0.0771	230	-48.300	<.0001
ps tv - s tv	-0.16	0.0771	230	-2.100	0.1720

```
P value adjustment: tukey method for comparing a family of 4 estimates
```

```
$TD
```

```
$emmeans
matrix microbe emmean    SE  df lower.CL upper.CL
ps      nv      29.7 2.530 230     24.8     34.7
s        nv      26.4 2.000 230     22.5     30.4
ps      tv      12.4 0.442 230     11.6     13.3
s        tv      11.3 0.363 230     10.6     12.0
```

Confidence level used: 0.95

```
$contrasts
contrast      estimate    SE  df t.ratio p.value
ps nv - s nv      3.30 3.22 230   1.030 0.7340
ps nv - ps tv     17.29 2.56 230   6.740 <.0001
ps nv - s tv     18.46 2.55 230   7.230 <.0001
s nv - ps tv     13.99 2.04 230   6.840 <.0001
s nv - s tv     15.16 2.03 230   7.470 <.0001
ps tv - s tv      1.17 0.57 230   2.040 0.1760
```

P value adjustment: tukey method for comparing a family of 4 estimates

```
## Interaction model (with matrices combined)
list(
  log10n0 = emmeans (m2_mod, pairwise ~ microbe, param = 'log10n0'),
  TD = emmeans (m2_mod, pairwise ~ microbe, param = 'TD'))
```

```
$log10n0
$emmeans
microbe emmean    SE  df lower.CL upper.CL
nv      3.92 0.0385 230     3.84     3.99
tv      7.61 0.0385 230     7.53     7.68
```

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
$contrasts
contrast estimate    SE  df t.ratio p.value
nv - tv      -3.69 0.0545 230 -67.700 <.0001
```

Results are averaged over the levels of: matrix

```
$TD
$emmeans
microbe emmean    SE  df lower.CL upper.CL
nv      28.1 1.610 230     24.9     31.2
tv      11.8 0.286 230     11.3     12.4
```

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
$contrasts
contrast estimate    SE  df t.ratio p.value
nv - tv           16.2 1.64 230   9.920 <.0001
```

Results are averaged over the levels of: matrix

```
# Joint tests
list(
  log10n0 = joint_tests (m2_mod, param = 'log10n0'),
  TD = joint_tests (m2_mod, param = 'TD'))
```

```
$log10n0
model term      df1 df2  F.ratio p.value
matrix          1 230    6.000  0.0180
microbe          1 230 4585.000 <.0001
matrix:microbe   1 230    0.000  0.5950
```

```
$TD
model term      df1 df2 F.ratio p.value
matrix          1 230   1.900  0.1730
microbe          1 230  98.500 <.0001
matrix:microbe   1 230   0.400  0.5140
```

```
# Overall marginal mean
emmeans (m2_mod, ~ microbe, param = 'log10n0')
```

```
microbe emmean    SE  df lower.CL upper.CL
nv       3.92 0.0385 230    3.84    3.99
tv       7.61 0.0385 230    7.53    7.68
```

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ matrix, param = 'TD')
```

```
matrix emmean    SE  df lower.CL upper.CL
ps      21.1 1.28 230    18.6    23.6
s       18.9 1.01 230    16.9    20.8
```

Results are averaged over the levels of: microbe
Confidence level used: 0.95

```
emmeans (m2_mod, ~ microbe, param = 'TD')
```

```
microbe emmean    SE  df lower.CL upper.CL
nv      28.1 1.610 230    24.9    31.2
tv      11.8 0.286 230    11.3    12.4
```

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
TD_values <- readRDS (here('obj2', 'results', "TD_values.RDS"))
TD_values$nv_pcr_1D = emmeans (m2_mod, ~ microbe, param = 'TD')[1]
TD_values$tv_pcr_1D = emmeans (m2_mod, ~ microbe, param = 'TD')[2]
saveRDS (TD_values, here('obj2', 'results', "TD_values.RDS"))
```

3.5.3 Assess covariance/collinearity

```
set.seed(SEED)
iter = ITER
```

```
## Variance-covariance matrix
m2_mod <- readRDS (here('obj2', 'results', 'nv_tv_pcr_mod_sum.RDS'))
m2_mod_coef <- coef (m2_mod)
```

```
(mod_vcov_joint <- vcov(m2_mod))
```

	log10n0.(Intercept)	log10n0.matrixs	
log10n0.microbetv			
log10n0.(Intercept)	0.00297	-0.00297	-
0.00297			
log10n0.matrixs	-0.00297	0.00594	
0.00297			
log10n0.microbetv	-0.00297	0.00297	
0.00594			
log10n0.matrixs:microbetv	0.00297	-0.00594	-
0.00594			
TD.(Intercept)	-0.11898	0.11898	
0.11898			
TD.matrixs	0.11898	-0.21299	-
0.11898			
TD.microbetv	0.11898	-0.11898	-
0.13980			
TD.matrixs:microbetv	-0.11898	0.21299	
0.13980			
	log10n0.matrixs:microbetv	TD.(Intercept)	TD.matrixs
log10n0.(Intercept)	0.00297	-0.119	0.119
log10n0.matrixs	-0.00594	0.119	-0.213
log10n0.microbetv	-0.00594	0.119	-0.119
log10n0.matrixs:microbetv	0.01188	-0.119	0.213
TD.(Intercept)	-0.11898	6.379	-6.379
TD.matrixs	0.21299	-6.379	10.362
TD.microbetv	0.13980	-6.379	6.379
TD.matrixs:microbetv	-0.25091	6.379	-10.362
	TD.microbetv	TD.matrixs:microbetv	
log10n0.(Intercept)	0.119	-0.119	
log10n0.matrixs	-0.119	0.213	
log10n0.microbetv	-0.140	0.140	
log10n0.matrixs:microbetv	0.140	-0.251	

TD.(Intercept)	-6.379	6.379
TD.matrixs	6.379	-10.362
TD.microbetv	6.575	-6.575
TD.matrixs:microbetv	-6.575	10.689

```

mod_vcov_indep <- diag(diag(mod_vcov_joint))
dimnames(mod_vcov_indep) <- dimnames(mod_vcov_joint)

## Draw of coefficients
iter = 5000
draw_joint <- MASS::mvrnorm(n=iter, mu = m2_mod_coef, Sigma = mod_vcov_joint)
|>
  as.data.frame()

## Assuming no covariance
draw_indep <- MASS::mvrnorm(n=iter, mu =m2_mod_coef, Sigma = mod_vcov_indep)
|>
  as.data.frame()

## Extract TD values
TD_joint <- cbind (ps_nv_joint =
  draw_joint[["TD.(Intercept)"]],
  s_nv_joint =
  draw_joint[["TD.(Intercept)"]] +
  draw_joint[["TD.matrixs"]],
  ps_tv_joint =
  draw_joint[["TD.(Intercept)"]] +
  draw_joint[["TD.microbetv"]],
  s_tv_joint =
  draw_joint[["TD.(Intercept)"]] +
  draw_joint[["TD.matrixs"]] +
  draw_joint[["TD.microbetv"]] +
  draw_joint[["TD.matrixs:microbetv"]]
)
colMeans(TD_joint <= 0)

ps_nv_joint    s_nv_joint    ps_tv_joint    s_tv_joint
      0           0           0           0

TD_indep <- cbind (ps_nv_indep =
  draw_indep[["TD.(Intercept)"]],
  s_nv_indep =
  draw_indep[["TD.(Intercept)"]] +
  draw_indep[["TD.matrixs"]],
  ps_tv_indep =
  draw_indep[["TD.(Intercept)"]] +
  draw_indep[["TD.microbetv"]],
  s_tv_indep =
  draw_indep[["TD.(Intercept)"]] +
  draw_indep[["TD.matrixs"]] +

```

```

        draw_indep[["TD.microbetv"]] +
        draw_indep[["TD.matrixs:microbetv"]]
    )

colMeans(TD_indep <= 0)

ps_nv_indep  s_nv_indep ps_tv_indep  s_tv_indep
      0.0000      0.0000      0.0000      0.0262

## Calculate mean and 95% QIs
QIs <- function(x) c(
  mean=mean(x),
  lwr=quantile(x, 0.025),
  upr=quantile(x, 0.975))

round(t(apply(TD_joint, 2, QIs)),2)

      mean lwr.2.5% upr.97.5%
ps_nv_joint 29.7      24.7      34.8
s_nv_joint  26.4      22.4      30.2
ps_tv_joint 12.4      11.6      13.3
s_tv_joint  11.3      10.6      12.0

round(t(apply(TD_indep, 2, QIs)),2)

      mean lwr.2.5% upr.97.5%
ps_nv_indep 29.7      24.76      34.7
s_nv_indep  26.4      18.53      34.3
ps_tv_indep 12.4       5.31      19.6
s_tv_indep  11.2      -0.09      22.8

## Agin compare with emmeans
emmeans (m2_mod, pairwise ~ matrix:microbe, param = 'TD')$emmeans

matrix microbe emmean    SE df lower.CL upper.CL
ps      nv      29.7 2.530 230      24.8      34.7
s      nv      26.4 2.000 230      22.5      30.4
ps      tv      12.4 0.442 230      11.6      13.3
s      tv      11.3 0.363 230      10.6      12.0

Confidence level used: 0.95

## If considering filtering out negative simulated numbers
TF_pos_joint <- rowSums(TD_joint <= 0 | !is.finite(TD_joint)) == 0
TF_pos_indep <- rowSums(TD_indep <= 0 | !is.finite(TD_indep)) == 0

mean(TF_pos_joint)

[1] 1

mean(TF_pos_indep)

```



```
[1] 0.974
```

```
CI <- function(x) c(mean = mean(x),  
                    lwr = unname(quantile(x, 0.025)),  
                    upr = unname(quantile(x, 0.975)))
```

```
as.data.frame(t(apply(TD_joint[TF_pos_joint, , drop = FALSE], 2, QIs)))
```

	mean	lwr.2.5%	upr.97.5%
ps_nv_joint	29.7	24.7	34.8
s_nv_joint	26.4	22.4	30.2
ps_tv_joint	12.4	11.6	13.3
s_tv_joint	11.3	10.6	12.0

```
as.data.frame(t(apply(TD_indep[TF_pos_indep, , drop = FALSE], 2, QIs)))
```

	mean	lwr.2.5%	upr.97.5%
ps_nv_indep	29.7	24.87	34.7
s_nv_indep	26.6	19.06	34.3
ps_tv_indep	12.5	5.69	19.6
s_tv_indep	11.6	1.52	22.8

```
saveRDS(mod_vcov_joint, here('obj2', 'results', 'nv_tv_pcr_sim_mvrnorm.RDS'))
```

```
# Remove selected objects from the environment
```

```
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),  
   envir = .GlobalEnv)
```

3.5.4 Plot microbial inactivation

```
m2 <- readRDS(here('obj2', 'data', 'Obj2_dat.RDS')) |>  
  dplyr::filter (assay=='pcr') |>  
  dplyr::mutate(matrix = factor(matrix,  
                                labels = c('Soil with plant', 'Soil'))) |>  
  droplevels()
```

```
m2_preds = readRDS(here('obj2', 'results', 'nv_tv_pcr_mod_preds.RDS'))  
m2_preds$matrix <- factor(m2_preds$matrix, levels = c("ps", "s"),  
                          labels = c("Soil with plant", "Soil"))
```

```
microb_labs = as_labeller(c(tv = 'Tulane virus', nv = "HuNoV GII"))
```

```
LOD1 <- LOD |>  
  dplyr::filter (microbe=='nv' | microbe=='tv')
```

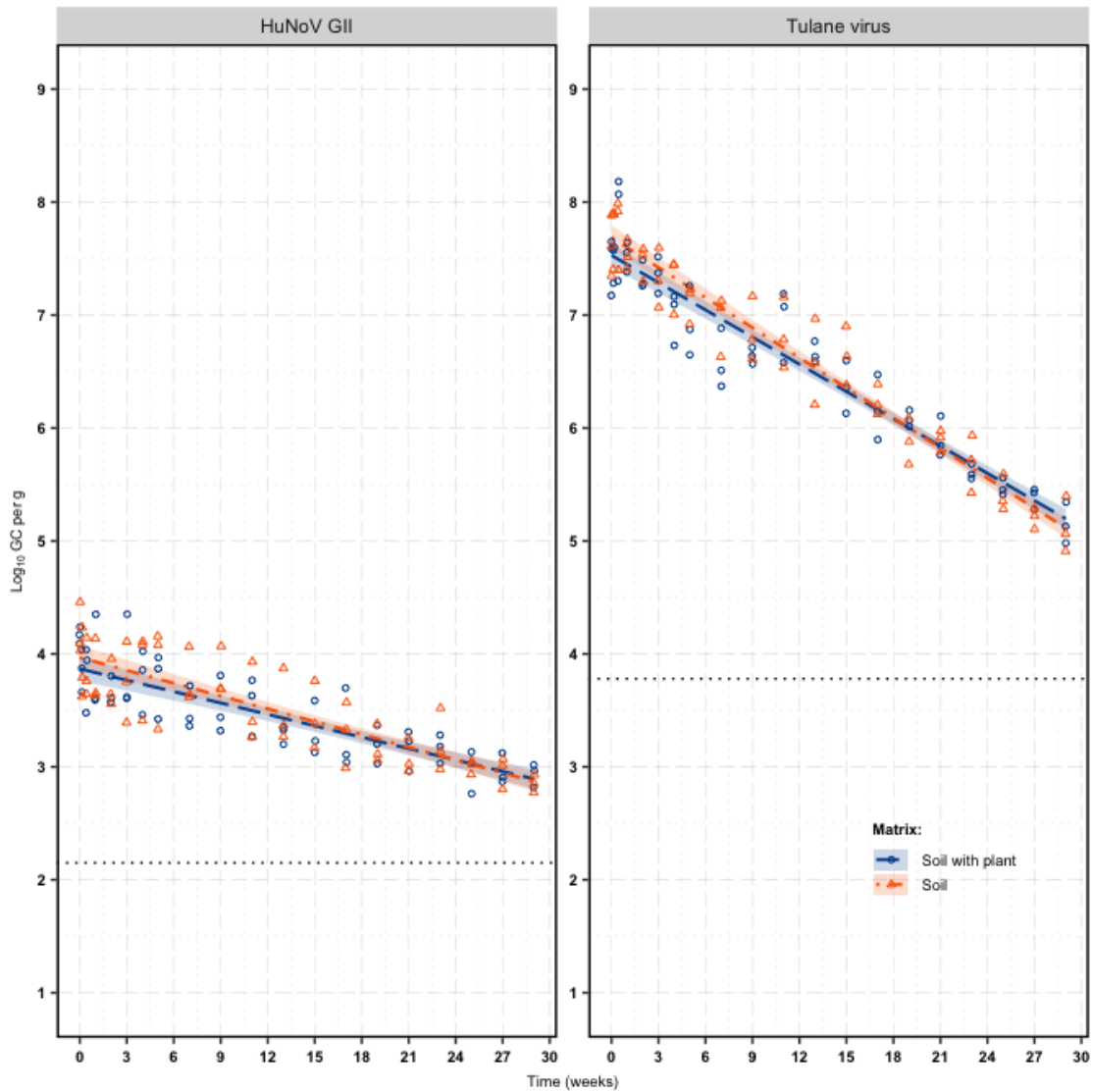
```
p_m2 = ggplot(m2_preds, aes(x = week, color = matrix, fill = matrix, linetype  
= matrix)) +  
  facet_wrap(~ microbe, labeller = microb_labs, scales = 'free_y') +  
  geom_hline(data = LOD1, aes(yintercept = lod, group = microbe),  
            color = 'black', linetype = 3) +  
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high,
```

```

      group = matrix),
      alpha = 0.2,
      color = NA) +
geom_line(aes(y = estimate), linewidth = 0.7) +
geom_point(data = m2,
  aes(y = survival, shape = matrix),
  size = 1,
  position = position_jitter(width = 0.03)) +
guides(
  fill = guide_legend(title = 'Matrix:'),
  color = guide_legend(title = 'Matrix:'),
  linetype = guide_legend(title = 'Matrix:'),
  shape = guide_legend(title = 'Matrix:')) +
labs(x = "Time (weeks)",
  y = bquote(Log[10]~GC~per~g)) +
scale_fill_manual(values = c("#00529b", "#fb6502")) +
scale_color_manual(values = c("#00529b", "#fb6502")) +
scale_shape_manual(values = c(1, 2)) +
scale_linetype_manual(values = c("longdash", "dotdash")) +
scale_x_continuous(
  breaks = seq(0, 30, 3),
  labels = seq(0, 30, 3)) +
scale_y_continuous(
  breaks = seq(1, 9, 1),
  labels = seq(1, 9, 1),
  limits = c(1,9)) +
theme(text = element_text(family = "Arial"),
  legend.position = 'inside', legend.position.inside = c(0.86, 0.18),
  legend.direction = "vertical",
  legend.text = element_text(size = 7),
  legend.title = element_text(size = 7, face = "bold"),
  legend.key.height = unit(0.4, "cm"),
  plot.title = element_text(color = "black", size = 7, face = "bold",
hjust = 0.5),
  axis.text.x = element_text(size = 7, color = "black", face = "bold",
hjust = 0.5, vjust = 0.5),
  axis.text.y = element_text(size = 7, color = "black", face = "bold"),
  axis.title.x = element_text(size = 7, vjust = -1),
  axis.title.y = element_text(size = 7, vjust = 2),
  panel.border = element_rect(color = "black", fill = NA, linewidth =
1),
  panel.grid.major = element_line(color = "gray", linewidth = 0.1,
linetype = "longdash"),
  panel.grid.minor = element_line(color = "lightgray", linewidth = 0.1,
linetype = "dotted"),
  panel.background = element_rect(fill = "transparent"),
  legend.background = element_rect(fill = "transparent", color =
"transparent"),
  plot.background = element_rect(fill = "transparent", color =
"transparent"))

```

p_m2



```
ggsave(file = here('obj2', 'results', 'Fig 2 tv_nv_pcr.tiff'),
        plot = p_m2,
        width = 12, height = 6, units = c("cm"), dpi=400)

# Remove selected objects from the environment
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
    envir = .GlobalEnv)
```

4. T2D

4.1 Model Functions

```
# Log linear and non-linear models

## Log-linear and reparameterized log-linear
llrepar <- function(x, log10n0, TD, D = 2){
  log10n0 - D*x/TD
}

## Nonlinear models
# Weibull and reparameterized Weibull
wbrepar <- function(x, log10n0, TD, beta, D = 2){
  log10n0 - D*(x/TD)^beta
}

# To clean the environment
list_exc <- c(lsf.str(envir = .GlobalEnv), "list_exc")
```

4.2 *E. coli*

```
m2 <- readRDS(here('obj2', 'data', 'obj2_dat.RDS')) |>
  dplyr::filter(microbe == 'ecoli',
                assay == "plate") |>
  droplevels()

m2_mod_wbrepar = nlme::gnls(survival ~ wbrepar(x = week,
                                              log10n0 = log10n0,
                                              TD = TD,
                                              beta = beta),
                           data = m2,
                           params = log10n0+TD+beta ~ matrix,
                           start = list(log10n0 = c(5, rep(0, 1)),
                                         TD = c(1, 0),
                                         beta = c(1, 0)))

(m2_mod <- m2_mod_wbrepar)

Generalized nonlinear least squares fit
Model: survival ~ wbrepar(x = week, log10n0 = log10n0, TD = TD, beta =
beta)
Data: m2
Log-likelihood: -76.9

Coefficients:
log10n0.(Intercept)    log10n0.matrixs    TD.(Intercept)
TD.matrixs
0.3220              7.2874              -0.2448              6.1146
```

beta.(Intercept)	beta.matrixs
0.6566	-0.0166

Degrees of freedom: 120 total; 114 residual
Residual standard error: 0.471

`summary (m2_mod)`

Generalized nonlinear least squares fit

Model: survival ~ wbrepar(x = week, log10n0 = log10n0, TD = TD, beta = beta)

Data: m2

AIC BIC logLik

168 187 -76.9

Coefficients:

	Value	Std.Error	t-value	p-value
log10n0.(Intercept)	7.29	0.181	40.2	0.000
log10n0.matrixs	-0.24	0.259	-0.9	0.346
TD.(Intercept)	6.11	1.040	5.9	0.000
TD.matrixs	0.32	1.549	0.2	0.836
beta.(Intercept)	0.66	0.062	10.6	0.000
beta.matrixs	-0.02	0.089	-0.2	0.852

Correlation:

	l100.(lg100.	TD.(I)	TD.mtr	bt.(I)
log10n0.matrixs	-0.701				
TD.(Intercept)	-0.910	0.638			
TD.matrixs	0.611	-0.912	-0.672		
beta.(Intercept)	-0.792	0.556	0.953	-0.640	
beta.matrixs	0.549	-0.794	-0.661	0.950	-0.694

Standardized residuals:

Min	Q1	Med	Q3	Max
-1.930	-0.745	-0.180	0.690	2.700

Residual standard error: 0.471

Degrees of freedom: 120 total; 114 residual

Variance-covariance matrix

`(mod_vcov <- vcov(m2_mod))`

	log10n0.(Intercept)	log10n0.matrixs	TD.(Intercept)
log10n0.(Intercept)	0.03293	-0.03293	-0.1717
log10n0.matrixs	-0.03293	0.06693	0.1717
TD.(Intercept)	-0.17174	0.17174	1.0823
TD.matrixs	0.17174	-0.36532	-1.0823
beta.(Intercept)	-0.00887	0.00887	0.0612
beta.matrixs	0.00887	-0.01827	-0.0612
	TD.matrixs	beta.(Intercept)	beta.matrixs

log10n0.(Intercept)	0.1717	-0.00887	0.00887
log10n0.matrixs	-0.3653	0.00887	-0.01827
TD.(Intercept)	-1.0823	0.06118	-0.06118
TD.matrixs	2.3980	-0.06118	0.13094
beta.(Intercept)	-0.0612	0.00381	-0.00381
beta.matrixs	0.1309	-0.00381	0.00792

Overall marginal means

```
emmeans (m2_mod, ~ 1, param = 'log10n0')
```

1	emmean	SE	df	lower.CL	upper.CL
overall	7.16	0.129	113	6.91	7.42

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ 1, param = 'beta')
```

1	emmean	SE	df	lower.CL	upper.CL
overall	0.648	0.0445	113	0.56	0.736

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ 1, param = 'TD')
```

1	emmean	SE	df	lower.CL	upper.CL
overall	6.28	0.774	113	4.74	7.81

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
TD_values <- readRDS (here('obj2', 'results', "TD_values.RDS"))
TD_values$ecoli_plate_2D = emmeans (m2_mod, ~ 1, param = 'TD')
saveRDS (TD_values, here('obj2', 'results', "TD_values.RDS"))
```

Remove selected objects from the environment

```
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
    envir = .GlobalEnv)
```

4.3 Tulane virus TCID50

```
m2 <- readRDS(here('obj2', 'data', 'Obj2_dat.RDS')) |>
  dplyr::filter (microbe=='tv', assay=='tcid50') |>
  droplevels()
```

```
m2_mod_wbepar = nlme::gnls(survival ~ wbepar(x = week,
                                             log10n0 = log10n0,
                                             TD = TD,
                                             beta = beta),
                           data = m2,
                           params = log10n0+TD+beta ~ matrix,
```

```

start = list(log10n0 = c(5, 0),
             TD = c(5, 0),
             beta = c(2, 0)))

(m2_mod <- m2_mod_wbrepair)

Generalized nonlinear least squares fit
Model: survival ~ wbrepair(x = week, log10n0 = log10n0, TD = TD, beta =
beta)
Data: m2
Log-likelihood: 24.2

Coefficients:
log10n0.(Intercept)      log10n0.matrixs      TD.(Intercept)
TD.matrixs
      4.7517              0.1390              13.2252      -
0.2451
      beta.(Intercept)      beta.matrixs
      0.8288              -0.0182

Degrees of freedom: 120 total; 114 residual
Residual standard error: 0.203

summary (m2_mod)

Generalized nonlinear least squares fit
Model: survival ~ wbrepair(x = week, log10n0 = log10n0, TD = TD, beta =
beta)
Data: m2
      AIC BIC logLik
-34.5 -15   24.2

Coefficients:
              Value Std.Error t-value p-value
log10n0.(Intercept)  4.75    0.067   70.5  0.000
log10n0.matrixs      0.14    0.096    1.4  0.150
TD.(Intercept)      13.23    0.710   18.6  0.000
TD.matrixs          -0.25    1.011   -0.2  0.809
beta.(Intercept)     0.83    0.046   17.8  0.000
beta.matrixs        -0.02    0.065   -0.3  0.781

Correlation:
              l100.( lg100. TD.(I) TD.mtr bt.(I)
log10n0.matrixs -0.702
TD.(Intercept)  -0.916  0.643
TD.matrixs      0.643 -0.917 -0.702
beta.(Intercept) -0.761  0.534  0.884 -0.621
beta.matrixs     0.543 -0.763 -0.631  0.885 -0.714

Standardized residuals:

```

Min	Q1	Med	Q3	Max
-2.9461	-0.6764	0.0461	0.6183	2.2847

Residual standard error: 0.203
Degrees of freedom: 120 total; 114 residual

```
# Variance-covariance matrix
(mod_vcov <- vcov(m2_mod))
```

	log10n0.(Intercept)	log10n0.matrixs	TD.(Intercept)
log10n0.(Intercept)	0.00455	-0.00455	-0.0439
log10n0.matrixs	-0.00455	0.00922	0.0439
TD.(Intercept)	-0.04386	0.04386	0.5047
TD.matrixs	0.04386	-0.08902	-0.5047
beta.(Intercept)	-0.00239	0.00239	0.0292
beta.matrixs	0.00239	-0.00477	-0.0292

	TD.matrixs	beta.(Intercept)	beta.matrixs
log10n0.(Intercept)	0.0439	-0.00239	0.00239
log10n0.matrixs	-0.0890	0.00239	-0.00477
TD.(Intercept)	-0.5047	0.02920	-0.02920
TD.matrixs	1.0229	-0.02920	0.05831
beta.(Intercept)	-0.0292	0.00216	-0.00216
beta.matrixs	0.0583	-0.00216	0.00424

```
# Overall marginal means
```

```
emmeans (m2_mod, ~ 1, param = 'log10n0')
```

1	emmean	SE	df	lower.CL	upper.CL
overall	4.82	0.048	113	4.73	4.92

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ 1, param = 'TD')
```

1	emmean	SE	df	lower.CL	upper.CL
overall	13.1	0.506	113	12.1	14.1

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ 1, param = 'beta')
```

1	emmean	SE	df	lower.CL	upper.CL
overall	0.82	0.0326	113	0.755	0.884

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
TD_values <- readRDS (here('obj2', 'results', "TD_values.RDS"))
TD_values$tv_tcid50_2D = emmeans (m2_mod, ~ 1, param = 'TD')
```



```
saveRDS (TD_values, here('obj2', 'results', "TD_values.RDS"))
```

```
# Remove selected objects from the environment
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
    envir = .GlobalEnv)
```

4.4 TuV and HuNoV - *RNase RT-qPCR*

```
m2 <- readRDS(here('obj2', 'data', 'Obj2_dat.RDS')) |>
  dplyr::filter (assay=='pcr') |>
  droplevels()
```

```
m2_mod_llrepar = nlme::gnls(survival ~ llrepar(x = week,
                                              log10n0 = log10n0,
                                              TD = TD),
                           data = m2,
                           params = log10n0+TD ~ matrix*microbe,
                           start = list(log10n0 = c(4, rep(0,3)),
                                         TD = c(25, rep(0,3))))
```

```
m2_mod_exp_wk = update(m2_mod_llrepar, weights = varExp(form = ~week))
```

```
(m2_mod <- m2_mod_exp_wk)
```

Generalized nonlinear least squares fit

Model: survival ~ llrepar(x = week, log10n0 = log10n0, TD = TD)

Data: m2

Log-likelihood: 20.4

Coefficients:

log10n0.(Intercept)	log10n0.matrixs	log10n0.microbetv
3.868	0.100	3.661
log10n0.matrixs:microbetv	TD.(Intercept)	TD.matrixs
0.058	59.456	-6.606
TD.microbetv	TD.matrixs:microbetv	
-34.584	4.271	

Variance function:

Structure: Exponential of variance covariate

Formula: ~week

Parameter estimates:

expon
-0.0259

Degrees of freedom: 240 total; 232 residual

Residual standard error: 0.305

```
summary (m2_mod)
```

Generalized nonlinear least squares fit

Model: survival ~ llrepar(x = week, log10n0 = log10n0, TD = TD)

```
Data: m2
      AIC BIC logLik
-20.8  14   20.4
```

```
Variance function:
  Structure: Exponential of variance covariate
  Formula: ~week
  Parameter estimates:
    expon
-0.0259
```

```
Coefficients:
               Value Std.Error t-value p-value
log10n0.(Intercept)    3.9     0.05   71.0  0.000
log10n0.matrixs        0.1     0.08    1.3  0.194
log10n0.microbetv       3.7     0.08   47.5  0.000
log10n0.matrixs:microbetv 0.1     0.11    0.5  0.595
TD.(Intercept)        59.5     5.05   11.8  0.000
TD.matrixs            -6.6     6.44   -1.0  0.306
TD.microbetv          -34.6     5.13   -6.7  0.000
TD.matrixs:microbetv    4.3     6.54    0.7  0.514
```

```
Correlation:
               l100.( lg10n0.mt lg10n0.mc l100.: TD.(I) TD.mtr
log10n0.matrixs    -0.707
log10n0.microbetv   -0.707  0.500
log10n0.matrixs:microbetv 0.500 -0.707  -0.707
TD.(Intercept)     -0.865  0.611   0.611  -0.432
TD.matrixs          0.678 -0.859  -0.480   0.607  -0.785
TD.microbetv         0.852 -0.602  -0.708   0.500  -0.985  0.773
TD.matrixs:microbetv -0.668  0.845   0.555  -0.704  0.773  -0.985
                  TD.mcr
log10n0.matrixs
log10n0.microbetv
log10n0.matrixs:microbetv
TD.(Intercept)
TD.matrixs
TD.microbetv
TD.matrixs:microbetv    -0.784
```

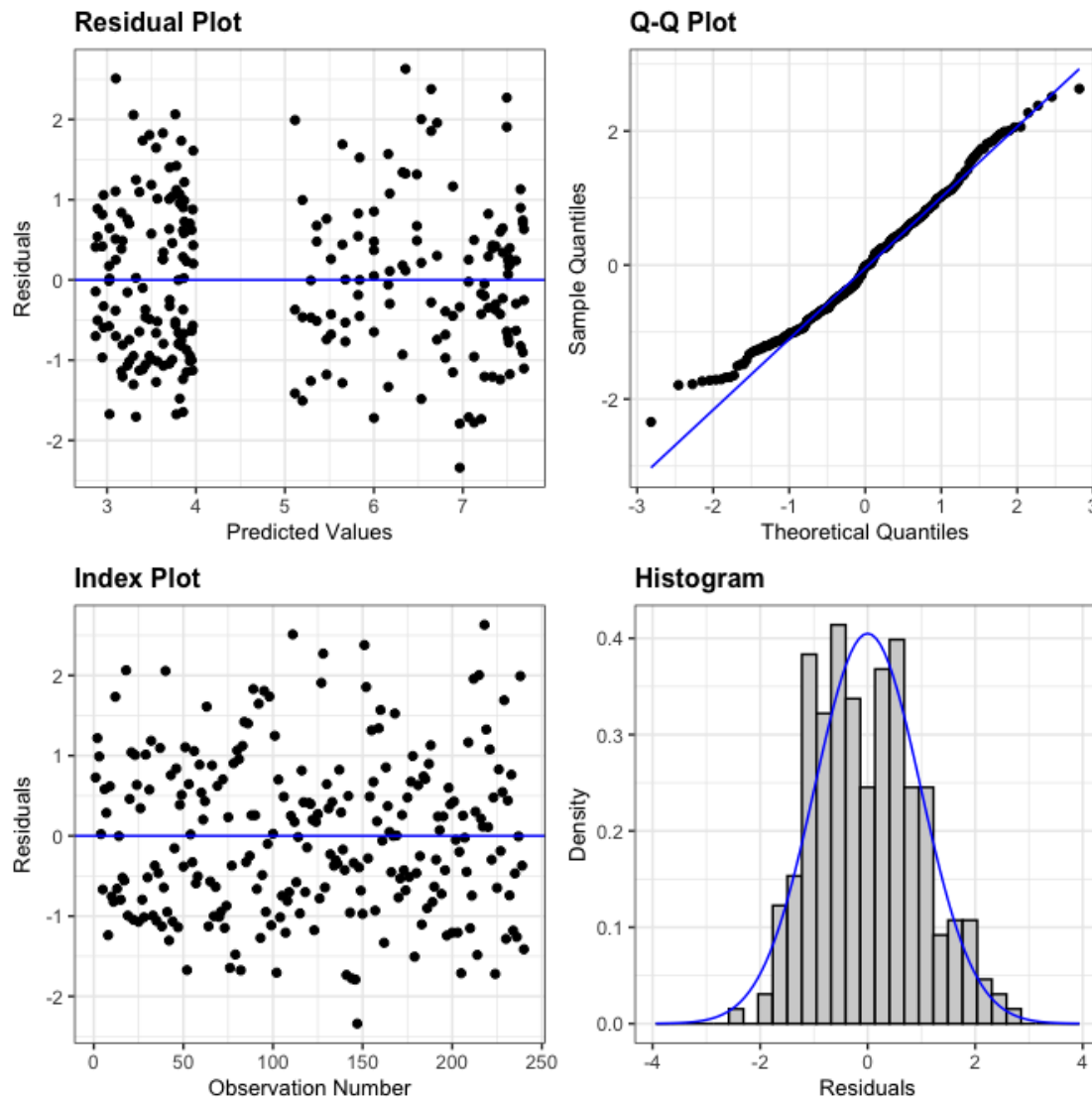
```
Standardized residuals:
      Min      Q1      Med      Q3      Max
-2.3405 -0.7486 -0.0182  0.6523  2.6306
```

```
Residual standard error: 0.305
Degrees of freedom: 240 total; 232 residual
```

```
# Residual plots
m2_res_m2_mod = resid(m2_mod, type = 'normalized')
```

```
m2_fit_m2_mod = fitted(m2_mod)
resid_auxpanel(m2_res_m2_mod, m2_fit_m2_mod)
```

Warning: Removed 1 row containing missing values or values outside the scale range (``geom_bar()``).



```
# Overall marginal mean
emmeans (m2_mod, ~ microbe, param = 'log10n0')
```

NOTE: Results may be misleading due to involvement in interactions

microbe	emmean	SE	df	lower.CL	upper.CL
nv	3.92	0.0385	230	3.84	3.99
tv	7.61	0.0385	230	7.53	7.68

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ matrix, param = 'TD')
```

NOTE: Results may be misleading due to involvement in interactions

	matrix	emmean	SE	df	lower.CL	upper.CL
ps		42.2	2.56	230	37.1	47.2
s		37.7	2.03	230	33.7	41.7

Results are averaged over the levels of: microbe
Confidence level used: 0.95

```
emmeans (m2_mod, ~ microbe, param = 'TD')
```

NOTE: Results may be misleading due to involvement in interactions

	microbe	emmean	SE	df	lower.CL	upper.CL
nv		56.2	3.22	230	49.8	62.5
tv		23.7	0.57	230	22.6	24.8

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
TD_values <- readRDS (here('obj2', 'results', "TD_values.RDS"))  
TD_values$nv_pcr_2D = emmeans (m2_mod, ~ microbe, param = 'TD')[1]
```

NOTE: Results may be misleading due to involvement in interactions

```
TD_values$tv_pcr_2D = emmeans (m2_mod, ~ microbe, param = 'TD')[2]
```

NOTE: Results may be misleading due to involvement in interactions

```
saveRDS (TD_values, here('obj2', 'results', "TD_values.RDS"))
```

```
# Remove selected objects from the environment  
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),  
    envir = .GlobalEnv)
```

5. T3D

5.1 Model Functions

```
# Log linear and non-linear models
```

```
## Log-linear and reparameterized log-linear  
llrepar <- function(x, log10n0, TD, D = 3){  
  log10n0 - D*x/TD  
}
```

```
## Nonlinear models
# Weibull and reparameterized Weibull
wbrepar <- function(x, log10n0, TD, beta, D = 3){
  log10n0 - D*(x/TD)^beta
}

# To clean the environment
list_exc <- c(lsf.str(envir = .GlobalEnv), "list_exc")
```

5.2 *E. coli*

```
m2 <- readRDS(here('obj2', 'data', 'obj2_dat.RDS')) |>
  dplyr::filter(microbe == 'ecoli',
                assay == "plate") |>
  droplevels()

m2_mod_wbrepar = nlme::gnls(survival ~ wbrepar(x = week,
                                              log10n0 = log10n0,
                                              TD = TD,
                                              beta = beta),
                           data = m2,
                           params = log10n0+TD+beta ~ matrix,
                           start = list(log10n0 = c(5, rep(0, 1)),
                                         TD = c(1, 0),
                                         beta = c(1, 0)))
```

```
(m2_mod <- m2_mod_wbrepar)
```

Generalized nonlinear least squares fit

Model: survival ~ wbrepar(x = week, log10n0 = log10n0, TD = TD, beta = beta)

Data: m2

Log-likelihood: -76.9

Coefficients:

log10n0.(Intercept)	log10n0.matrixs	TD.(Intercept)
7.2874	-0.2448	11.3389
0.7902		
beta.(Intercept)	beta.matrixs	
0.6566	-0.0166	

Degrees of freedom: 120 total; 114 residual

Residual standard error: 0.471

```
summary (m2_mod)
```

Generalized nonlinear least squares fit

Model: survival ~ wbrepar(x = week, log10n0 = log10n0, TD = TD, beta = beta)

Data: m2

```
AIC BIC logLik
168 187 -76.9
```

Coefficients:

	Value	Std.Error	t-value	p-value
log10n0.(Intercept)	7.29	0.181	40.2	0.000
log10n0.matrixs	-0.24	0.259	-0.9	0.346
TD.(Intercept)	11.34	1.317	8.6	0.000
TD.matrixs	0.79	1.960	0.4	0.688
beta.(Intercept)	0.66	0.062	10.6	0.000
beta.matrixs	-0.02	0.089	-0.2	0.852

Correlation:

	1100.(lg100.	TD.(I)	TD.mtr	bt.(I)
log10n0.matrixs	-0.701				
TD.(Intercept)	-0.936	0.657			
TD.matrixs	0.629	-0.938	-0.672		
beta.(Intercept)	-0.792	0.556	0.895	-0.602	
beta.matrixs	0.549	-0.794	-0.621	0.888	-0.694

Standardized residuals:

	Min	Q1	Med	Q3	Max
	-1.930	-0.745	-0.180	0.690	2.700

Residual standard error: 0.471

Degrees of freedom: 120 total; 114 residual

```
# Variance-covariance matrix
(mod_vcov <- vcov(m2_mod))
```

	log10n0.(Intercept)	log10n0.matrixs	TD.(Intercept)
log10n0.(Intercept)	0.03293	-0.03293	-0.2238
log10n0.matrixs	-0.03293	0.06693	0.2238
TD.(Intercept)	-0.22383	0.22383	1.7353
TD.matrixs	0.22383	-0.47576	-1.7353
beta.(Intercept)	-0.00887	0.00887	0.0728
beta.matrixs	0.00887	-0.01827	-0.0728

	TD.matrixs	beta.(Intercept)	beta.matrixs
log10n0.(Intercept)	0.2238	-0.00887	0.00887
log10n0.matrixs	-0.4758	0.00887	-0.01827
TD.(Intercept)	-1.7353	0.07281	-0.07281
TD.matrixs	3.8430	-0.07281	0.15489
beta.(Intercept)	-0.0728	0.00381	-0.00381
beta.matrixs	0.1549	-0.00381	0.00792

```
# Overall marginal means
```

```
emmeans (m2_mod, ~ 1, param = 'log10n0')
```

	1	emmean	SE	df	lower.CL	upper.CL
overall		7.16	0.129	113	6.91	7.42

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ 1, param = 'beta')

1      emmean    SE  df lower.CL upper.CL
overall 0.648 0.0445 113    0.56    0.736
```

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ 1, param = 'TD')

1      emmean    SE  df lower.CL upper.CL
overall 11.7 0.98 113    9.79    13.7
```

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
TD_values <- readRDS (here('obj2', 'results', "TD_values.RDS"))
TD_values$ecoli_plate_3D = emmeans (m2_mod, ~ 1, param = 'TD')
saveRDS (TD_values, here('obj2', 'results', "TD_values.RDS"))

# Remove selected objects from the environment
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
    envir = .GlobalEnv)
```

5.3 Tulane virus TCID50

```
m2 <- readRDS(here('obj2', 'data', 'Obj2_dat.RDS')) |>
  dplyr::filter (microbe=='tv', assay=='tcid50') |>
  droplevels()
```

```
m2_mod_wbrep = nlme::gnls(survival ~ wbrep(x = week,
                                         log10n0 = log10n0,
                                         TD = TD,
                                         beta = beta),
  data = m2,
  params = log10n0+TD+beta ~ matrix,
  start = list(log10n0 = c(5, 0),
               TD = c(5, 0),
               beta = c(2, 0)))
```

```
(m2_mod <- m2_mod_wbrep)
```

Generalized nonlinear least squares fit

Model: survival ~ wbrep(x = week, log10n0 = log10n0, TD = TD, beta = beta)

Data: m2

Log-likelihood: 24.2

```

Coefficients:
log10n0.(Intercept)    log10n0.matrixs    TD.(Intercept)
TD.matrixs
      4.7517              0.1390              21.5708      -
0.1663
      beta.(Intercept)    beta.matrixs
      0.8288              -0.0182

```

Degrees of freedom: 120 total; 114 residual
Residual standard error: 0.203

`summary (m2_mod)`

Generalized nonlinear least squares fit

Model: survival ~ wbrepar(x = week, log10n0 = log10n0, TD = TD, beta = beta)

Data: m2

AIC BIC logLik
-34.5 -15 24.2

```

Coefficients:
              Value Std.Error t-value p-value
log10n0.(Intercept)  4.75    0.067   70.5  0.000
log10n0.matrixs      0.14    0.096    1.4  0.150
TD.(Intercept)      21.57    0.693   31.1  0.000
TD.matrixs          -0.17    0.992   -0.2  0.867
beta.(Intercept)     0.83    0.046   17.8  0.000
beta.matrixs         -0.02    0.065   -0.3  0.781

```

Correlation:

```

              1100.( lg100. TD.(I) TD.mtr bt.(I)
log10n0.matrixs -0.702
TD.(Intercept)  -0.881  0.619
TD.matrixs      0.615 -0.883 -0.698
beta.(Intercept) -0.761  0.534  0.624 -0.436
beta.matrixs     0.543 -0.763 -0.446  0.629 -0.714

```

Standardized residuals:

```

      Min      Q1      Med      Q3      Max
-2.9461 -0.6764  0.0461  0.6183  2.2847

```

Residual standard error: 0.203
Degrees of freedom: 120 total; 114 residual

Variance-covariance matrix
(mod_vcov <- vcov(m2_mod))

```

              log10n0.(Intercept) log10n0.matrixs TD.(Intercept)
log10n0.(Intercept)      0.00455      -0.00455      -0.0412

```


log10n0.matrixs	-0.00455	0.00922	0.0412
TD.(Intercept)	-0.04116	0.04116	0.4801
TD.matrixs	0.04116	-0.08414	-0.4801
beta.(Intercept)	-0.00239	0.00239	0.0201
beta.matrixs	0.00239	-0.00477	-0.0201

	TD.matrixs	beta.(Intercept)	beta.matrixs
log10n0.(Intercept)	0.0412	-0.00239	0.00239
log10n0.matrixs	-0.0841	0.00239	-0.00477
TD.(Intercept)	-0.4801	0.02011	-0.02011
TD.matrixs	0.9843	-0.02011	0.04064
beta.(Intercept)	-0.0201	0.00216	-0.00216
beta.matrixs	0.0406	-0.00216	0.00424

Overall marginal means

```
emmeans (m2_mod, ~ 1, param = 'log10n0')
```

1	emmean	SE	df	lower.CL	upper.CL
overall	4.82	0.048	113	4.73	4.92

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ 1, param = 'TD')
```

1	emmean	SE	df	lower.CL	upper.CL
overall	21.5	0.496	113	20.5	22.5

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ 1, param = 'beta')
```

1	emmean	SE	df	lower.CL	upper.CL
overall	0.82	0.0326	113	0.755	0.884

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
TD_values <- readRDS (here('obj2', 'results', "TD_values.RDS"))
TD_values$tv_tcid50_3D = emmeans (m2_mod, ~ 1, param = 'TD')
saveRDS (TD_values, here('obj2', 'results', "TD_values.RDS"))
```

Remove selected objects from the environment

```
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
   envir = .GlobalEnv)
```

5.4 TuV and HuNoV - RNase RT-qPCR

```
m2 <- readRDS(here('obj2', 'data', 'Obj2_dat.RDS')) |>
  dplyr::filter (assay=='pcr') |>
  droplevels()
```

```

m2_mod_llrepar = nlme::gnls(survival ~ llrepar(x = week,
                                             log10n0 = log10n0,
                                             TD = TD),
                           data = m2,
                           params = log10n0+TD ~ matrix*microbe,
                           start = list(log10n0 = c(4, rep(0,3)),
                                         TD = c(25, rep(0,3))))

m2_mod_exp_wk = update(m2_mod_llrepar, weights = varExp(form = ~week))

(m2_mod <- m2_mod_exp_wk)

Generalized nonlinear least squares fit
Model: survival ~ llrepar(x = week, log10n0 = log10n0, TD = TD)
Data: m2
Log-likelihood: 20.4

Coefficients:
      log10n0.(Intercept)      log10n0.matrixs      log10n0.microbetv
                3.868                0.100                3.661
log10n0.matrixs:microbetv      TD.(Intercept)      TD.matrixs
                0.058                89.183            -9.909
      TD.microbetv      TD.matrixs:microbetv
        -51.876                6.406

Variance function:
Structure: Exponential of variance covariate
Formula: ~week
Parameter estimates:
  expon
-0.0259
Degrees of freedom: 240 total; 232 residual
Residual standard error: 0.305

summary (m2_mod)

Generalized nonlinear least squares fit
Model: survival ~ llrepar(x = week, log10n0 = log10n0, TD = TD)
Data: m2
      AIC BIC logLik
-20.8  14   20.4

Variance function:
Structure: Exponential of variance covariate
Formula: ~week
Parameter estimates:
  expon
-0.0259

Coefficients:

```

	Value	Std.Error	t-value	p-value
log10n0.(Intercept)	3.9	0.05	71.0	0.000
log10n0.matrixs	0.1	0.08	1.3	0.194
log10n0.microbetv	3.7	0.08	47.5	0.000
log10n0.matrixs:microbetv	0.1	0.11	0.5	0.595
TD.(Intercept)	89.2	7.58	11.8	0.000
TD.matrixs	-9.9	9.66	-1.0	0.306
TD.microbetv	-51.9	7.69	-6.7	0.000
TD.matrixs:microbetv	6.4	9.81	0.7	0.514

Correlation:

	l100.(lg10n0.mt	lg10n0.mc	l100.:	TD.(I)	TD.mtr
log10n0.matrixs	-0.707					
log10n0.microbetv	-0.707	0.500				
log10n0.matrixs:microbetv	0.500	-0.707	-0.707			
TD.(Intercept)	-0.865	0.611	0.611	-0.432		
TD.matrixs	0.678	-0.859	-0.480	0.607	-0.785	
TD.microbetv	0.852	-0.602	-0.708	0.500	-0.985	0.773
TD.matrixs:microbetv	-0.668	0.845	0.555	-0.704	0.773	-0.985
	TD.mcr					

log10n0.matrixs
log10n0.microbetv
log10n0.matrixs:microbetv
TD.(Intercept)
TD.matrixs
TD.microbetv
TD.matrixs:microbetv -0.784

Standardized residuals:

Min	Q1	Med	Q3	Max
-2.3405	-0.7486	-0.0182	0.6523	2.6306

Residual standard error: 0.305

Degrees of freedom: 240 total; 232 residual

Residual plots

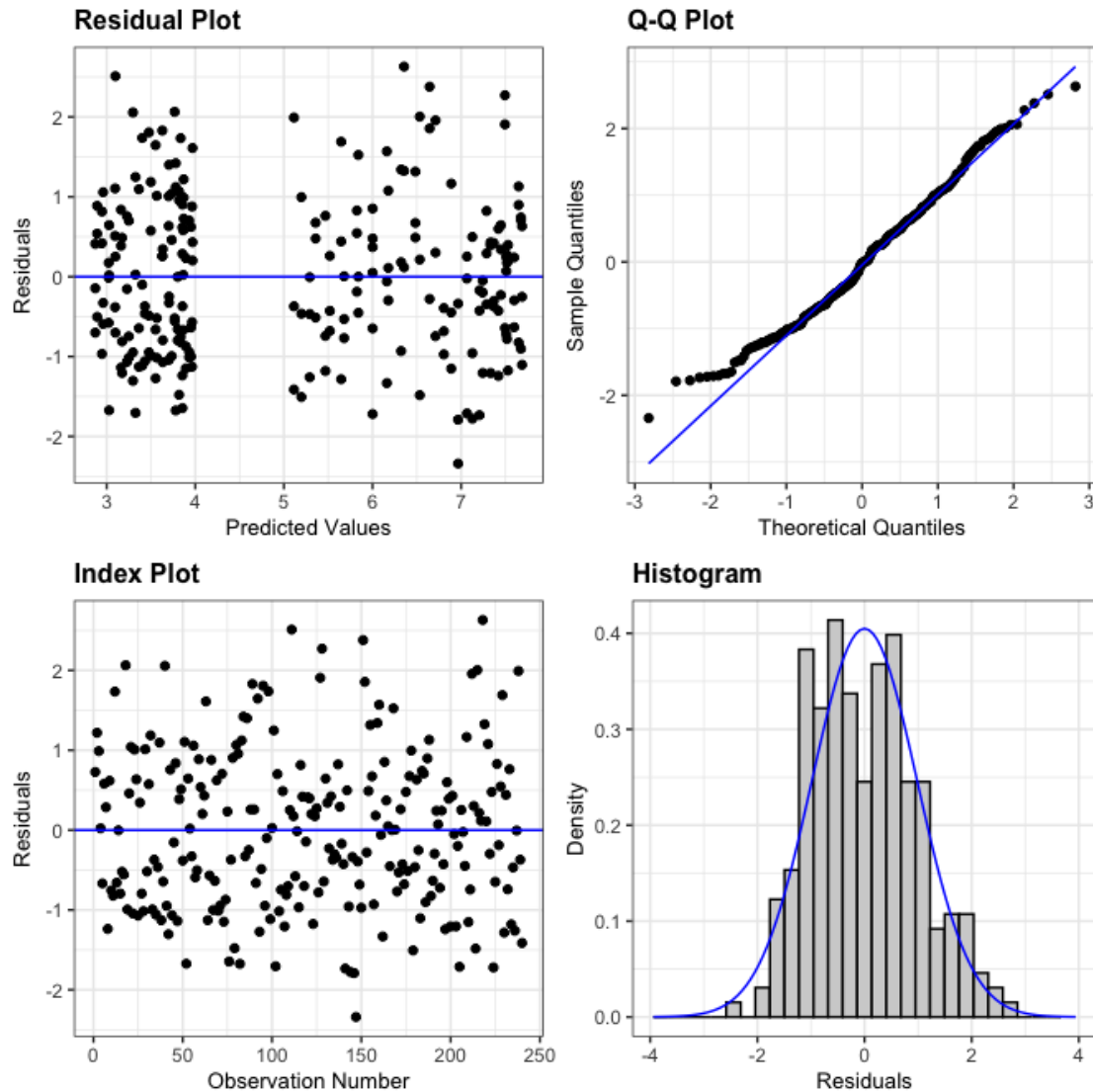
m2_res_m2_mod = resid(m2_mod, type = 'normalized')

m2_fit_m2_mod = fitted(m2_mod)

resid_auxpanel(m2_res_m2_mod, m2_fit_m2_mod)

Warning: Removed 2 rows containing missing values or values outside the scale range

(`geom_bar()`).



```
# Overall marginal mean
emmeans (m2_mod, ~ microbe, param = 'log10n0')
```

NOTE: Results may be misleading due to involvement in interactions

microbe	emmean	SE	df	lower.CL	upper.CL
nv	3.92	0.0385	230	3.84	3.99
tv	7.61	0.0385	230	7.53	7.68

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
emmeans (m2_mod, ~ matrix, param = 'TD')
```

NOTE: Results may be misleading due to involvement in interactions

matrix	emmean	SE	df	lower.CL	upper.CL
ps	63.2	3.85	230	55.7	70.8

```
s          56.5 3.04 230      50.5      62.5
```

Results are averaged over the levels of: microbe
Confidence level used: 0.95

```
emmeans (m2_mod, ~ microbe, param = 'TD')
```

NOTE: Results may be misleading due to involvement in interactions

microbe	emmean	SE	df	lower.CL	upper.CL
nv	84.2	4.83	230	74.7	93.7
tv	35.6	0.86	230	33.9	37.2

Results are averaged over the levels of: matrix
Confidence level used: 0.95

```
TD_values <- readRDS (here('obj2', 'results', "TD_values.RDS"))
TD_values$nv_pcr_3D = emmeans (m2_mod, ~ microbe, param = 'TD')[1]
```

NOTE: Results may be misleading due to involvement in interactions

```
TD_values$tv_pcr_3D = emmeans (m2_mod, ~ microbe, param = 'TD')[2]
```

NOTE: Results may be misleading due to involvement in interactions

```
saveRDS (TD_values, here('obj2', 'results', "TD_values.RDS"))
```

```
# Remove selected objects from the environment
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
    envir = .GlobalEnv)
```

6. Combined TD values

```
TD_values <- readRDS(here("obj2", "results", "TD_values.RDS"))
```

```
TD_df1 <- imap_dfr(TD_values, function(x, id) {
  TD_sum <- as.data.frame(summary(x))
```

```
  TD_sum <- TD_sum |>
  dplyr::mutate(ID = factor(id)) |>
  dplyr::select(ID, emmean, lower.CL, upper.CL)
```

```
    return(TD_sum)
})
```

```
TD_df2 <- TD_df1 |>
  separate(ID, into = c("microbe", "assay", "TnD"), sep = "_") %>%
  mutate(
    microbe = factor(microbe),
    TnD = factor(TnD, levels = c("1D", "2D", "3D")),
    assay = factor(assay)
```

```

)

TD_df2 <- TD_df2 |>
  dplyr::mutate(
    microbe = recode(microbe,
      "tv" = "TuV",
      "nv" = "HuNoV",
      "ecoli" = "E. coli"),
    assay = recode(assay,
      "pcr" = "PCR",
      "tcid50" = "TCID50",
      "plate" = "Plate Count")
  )

nv_tv <- TD_df2 |>
  filter (assay == "PCR") |>
  ggplot(aes(x = TnD, y = emmean)) +
  geom_point(size = 1.1) +
  facet_wrap ( . ~ microbe:assay) +
  geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL),
    width = 0.2) +
  scale_y_continuous(
    breaks = seq(10, 100, 15),
    labels = seq(10, 100, 15),
    limits = c(10, 100)) +
  labs(
    x = "First decimal reduction",
    y = "Weeks") +
  theme_bw()

ec_tv <- TD_df2 |>
  filter (assay != "PCR") |>
  ggplot(aes(x = TnD, y = emmean)) +
  geom_point(size = 1.1) +
  facet_wrap ( . ~ microbe:assay) +
  geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL),
    width = 0.2) +
  scale_y_continuous(
    breaks = seq(0, 25, 5),
    labels = seq(0, 25, 5),
    limits = c(0, 25)) +
  labs(
    x = "",
    y = "Weeks") +
  theme_bw()

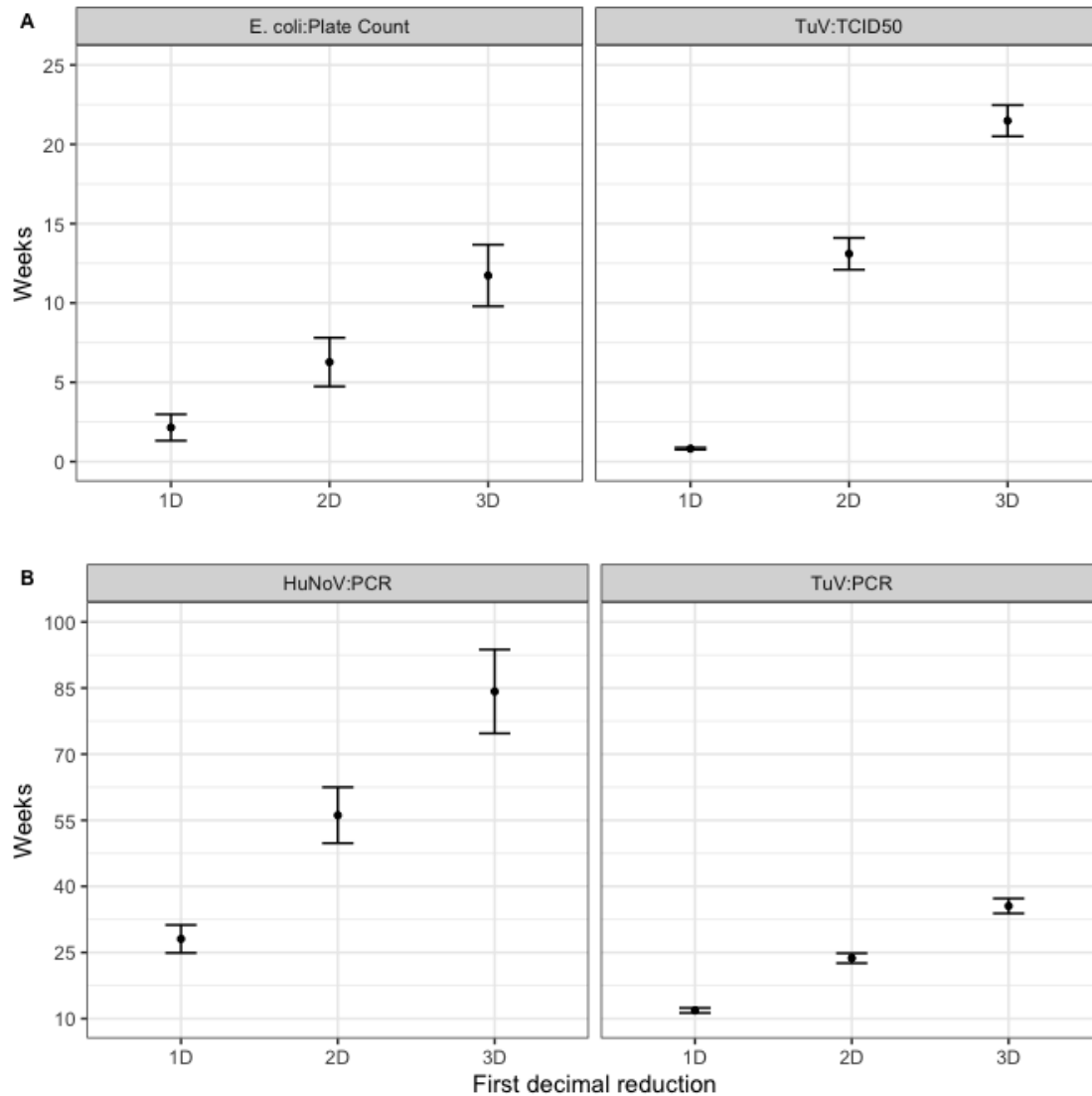
combined_plots <- ggpubr::ggarrange (ec_tv, nv_tv,
  labels = c("A", "B"),
  font.label = list(size = 9, face =

```

```
"bold"),
```

```
label.x = 0.01,  
label.y = 0.98,  
nrow=2, ncol=1)
```

```
combined_plots
```



```
#ggsave(filename = here('obj2', 'results', 'Fig S1 TnD_plot.tiff'), plot =  
combined_plots, width = 11, height = 10, units = "cm", dpi = 600)
```

```
TD_df1 |>  
  mutate(  
    Estimate_CI = sprintf("%.2f [%.2f, %.2f]", emmean, lower.CL, upper.CL)  
  ) |>  
  dplyr::select(ID, Estimate_CI)
```

	ID	Estimate	CI
1	ecoli_plate_1D	2.15	[1.32, 2.98]
2	tv_tcid50_1D	0.82	[0.76, 0.88]
3	nv_pcr_1D	28.08	[24.91, 31.25]
4	tv_pcr_1D	11.85	[11.29, 12.42]
5	ecoli_plate_2D	6.28	[4.74, 7.81]
6	tv_tcid50_2D	13.10	[12.10, 14.10]
7	nv_pcr_2D	56.15	[49.81, 62.50]
8	tv_pcr_2D	23.70	[22.58, 24.83]
9	ecoli_plate_3D	11.73	[9.79, 13.68]
10	tv_tcid50_3D	21.49	[20.50, 22.47]
11	nv_pcr_3D	84.23	[74.72, 93.74]
12	tv_pcr_3D	35.56	[33.87, 37.25]

```
# Remove selected objects from the environment
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
    envir = .GlobalEnv)
```

7. Correlation matrix

```
# Reshape data
m2 <- readRDS(here('obj2', 'data', 'Obj2_dat.RDS')) |>
  dplyr::filter (microbe=="nv" | microbe=="tv") |>
  dplyr::mutate(id = paste(microbe, assay, sep = "_")) |>
  dplyr::mutate(id = factor(id)) |>
  dplyr::select (survival, week, id) |>
  droplevels()

m2_wide <- m2 |>
  group_by(id, week) |>
  mutate(row = row_number()) |>
  pivot_wider(names_from = id, values_from = survival) |>
  as.data.frame() |>
  ungroup()

# rename columns
label_ids <- c(
  "HuNoV_PCR" = "nv_pcr",
  "TuV_PCR" = "tv_pcr",
  "TuV_TCID50" = "tv_tcid50")

m2_wide <- m2_wide |>
  dplyr::select(-week, -row) |>
  rename(!!!label_ids)

# Correlation matrix
# Custom function to show both r and p-values
cor_fun <- function(data, mapping, ...){
  x <- eval_data_col(data, mapping$x)
```



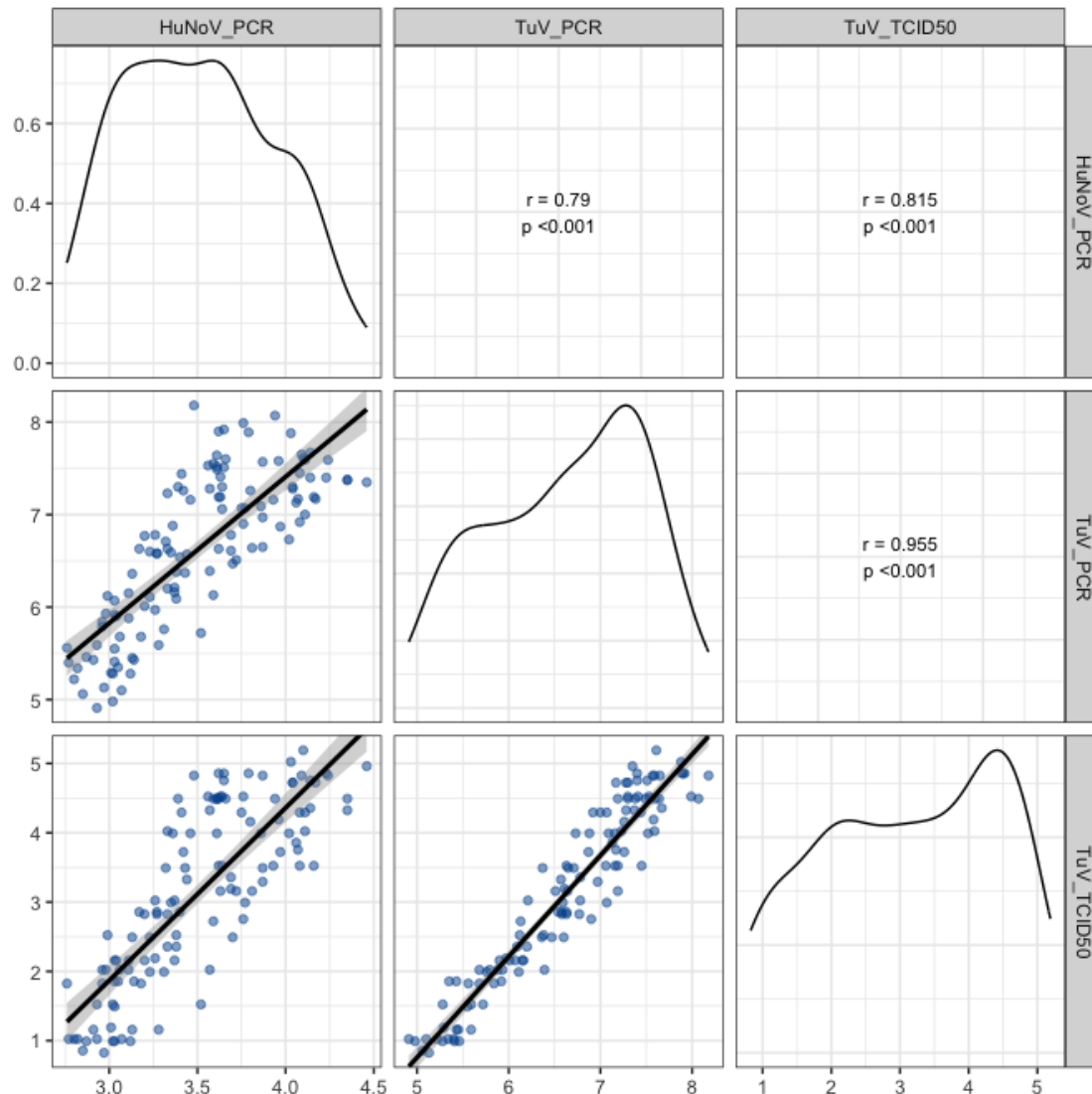
```

y <- eval_data_col(data, mapping$y)
cor_test <- cor.test(x, y)
r <- round(cor_test$estimate, 3)
p <- cor_test$p.value
txt <- paste0("r = ", r, "\n",
              "p ", ifelse(p < 0.001, "<0.001", round(p, 3)))
ggplot(data = data, mapping = mapping) +
  annotate("text", x = 0.5, y = 0.5, label = txt, size = 3, hjust = 0.5) +
  theme_void()
}

p_m2 <- GGally::ggpairs (m2_wide,
  lower = list(continuous = wrap("smooth",
                                method = "lm",
                                alpha=0.5,
                                se = TRUE,
                                color = "#00529b")),
  upper = list(continuous = cor_fun),
  diag = list(continuous = wrap("densityDiag",
                                color = "#000"))
) + theme_bw()

p_m2

```



```
ggsave(file = here('obj2', 'results', 'Fig 3 cor_matrix.tiff'),
       plot = p_m2,
       width=12, height=12, units = c("cm"), dpi=400)
```

```
# Correlations with 95% CI
cor.test(m2_wide$TuV_PCR,
        m2_wide$HuNoV_PCR,
        use = "pairwise.complete.obs")
```

Pearson's product-moment correlation

data: m2_wide\$TuV_PCR and m2_wide\$HuNoV_PCR
 t = 14, df = 118, p-value <2e-16
 alternative hypothesis: true correlation is not equal to 0
 95 percent confidence interval:

```
0.712 0.849
sample estimates:
cor
0.79
```

```
cor.test(m2_wide$TuV_PCR,
         m2_wide$TuV_TCID50,
         use = "pairwise.complete.obs")
```

Pearson's product-moment correlation

```
data: m2_wide$TuV_PCR and m2_wide$TuV_TCID50
t = 35, df = 118, p-value <2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.936 0.969
sample estimates:
cor
0.955
```

```
cor.test(m2_wide$HuNoV_PCR,
         m2_wide$TuV_TCID50,
         use = "pairwise.complete.obs")
```

Pearson's product-moment correlation

```
data: m2_wide$HuNoV_PCR and m2_wide$TuV_TCID50
t = 15, df = 118, p-value <2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.745 0.868
sample estimates:
cor
0.815
```

```
# Remove selected objects from the environment
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
    envir = .GlobalEnv)
```

8. PCR:Inf ratio for Tulane virus

```
readRDS(here('obj2', 'data', 'Obj2_dat.RDS')) |>
  dplyr::filter(microbe=='tv') |>
  droplevels() |>
  dplyr::select(matrix, week, replicate, assay, survival) |>
  tidyr::pivot_wider(names_from = assay, values_from = survival) |>
  dplyr::mutate(gc2inf = pcr-tcid50,
               week_fac = as.factor(week)) |>
```

```

saveRDS(here('obj2', 'results', 'gc2inf.RDS'))

m2 <- readRDS(here('obj2', 'results', 'gc2inf.RDS'))

# Ratio across matrices
lm2 <- nlme::gls(gc2inf ~ matrix * week_fac,
                weights = varIdent(form = ~1 | week_fac),
                data = m2)

# Overall significant differences
joint_tests(lm2, by = NULL)

  model term      df1 df2 F.ratio p.value
matrix      1  49   1.300  0.2630
week_fac    19   6  37.300  <.0001
matrix:week_fac 19   6   1.100  0.4960

joint_tests(lm2, by = "week_fac")

week_fac = 0:
  model term df1   df2 F.ratio p.value
matrix      1 12.00   0.420  0.5300

week_fac = 0.142857142857143:
  model term df1   df2 F.ratio p.value
matrix      1  3.92   0.090  0.7770

week_fac = 0.428571428571429:
  model term df1   df2 F.ratio p.value
matrix      1  3.99   0.080  0.7870

week_fac = 1:
  model term df1   df2 F.ratio p.value
matrix      1  2.63   0.060  0.8210

week_fac = 2:
  model term df1   df2 F.ratio p.value
matrix      1  3.13   0.050  0.8430

week_fac = 3:
  model term df1   df2 F.ratio p.value
matrix      1  3.99   0.010  0.9160

week_fac = 4:
  model term df1   df2 F.ratio p.value
matrix      1  4.00   0.460  0.5330

week_fac = 5:
  model term df1   df2 F.ratio p.value
matrix      1  3.98   0.120  0.7450

```

```
week_fac = 7:
  model term df1    df2 F.ratio p.value
matrix      1  3.00   1.770  0.2750
```

```
week_fac = 9:
  model term df1    df2 F.ratio p.value
matrix      1  4.00   1.570  0.2780
```

```
week_fac = 11:
  model term df1    df2 F.ratio p.value
matrix      1  3.99   6.040  0.0700
```

```
week_fac = 13:
  model term df1    df2 F.ratio p.value
matrix      1  2.66   1.620  0.3030
```

```
week_fac = 15:
  model term df1    df2 F.ratio p.value
matrix      1  3.94   0.310  0.6080
```

```
week_fac = 17:
  model term df1    df2 F.ratio p.value
matrix      1  3.96   0.020  0.9010
```

```
week_fac = 19:
  model term df1    df2 F.ratio p.value
matrix      1  3.99   5.650  0.0760
```

```
week_fac = 21:
  model term df1    df2 F.ratio p.value
matrix      1  3.97   0.460  0.5330
```

```
week_fac = 23:
  model term df1    df2 F.ratio p.value
matrix      1  3.90   0.810  0.4190
```

```
week_fac = 25:
  model term df1    df2 F.ratio p.value
matrix      1  3.96   1.980  0.2330
```

```
week_fac = 27:
  model term df1    df2 F.ratio p.value
matrix      1  4.00   8.980  0.0400
```

```
week_fac = 29:
  model term df1    df2 F.ratio p.value
matrix      1  2.82   0.070  0.8090
```

```

joint_tests(lm2, by = "matrix")

matrix = ps:
  model term df1 df2 F.ratio p.value
  week_fac    19 6.01 23.070 0.0004

matrix = s:
  model term df1 df2 F.ratio p.value
  week_fac    19 6.01 15.290 0.0014

# Pairwise comparisons between matrices at each week level
# emmeans(lm2, ~ matrix | week_fac, mode = "df.error") |>
# contrast(method = "pairwise", adjust = "none", infer = TRUE)

# Ratio across matrices at each week
emmeans(lm2, ~ week_fac | matrix, mode = "df.error") |>
  multcomp::cld(decreasing = TRUE, Letters = letters) |>
  arrange(week_fac)

matrix = ps:
  week_fac      emmean      SE df lower.CL upper.CL .group
0            2.68 0.1350 61      2.41      2.95      g
0.142857142857143 2.80 0.1630 61      2.48      3.13      fg
0.428571428571429 3.17 0.2720 61      2.63      3.71     bcdefg
1            3.03 0.1310 61      2.77      3.29      efg
2            3.02 0.1300 61      2.76      3.28      efg
3            3.09 0.2500 61      2.59      3.59      cdefg
4            3.00 0.2690 61      2.46      3.54      cdefg
5            3.28 0.2310 61      2.82      3.74     bcdefg
7            3.04 0.1420 61      2.76      3.32      efg
9            3.32 0.0592 61      3.20      3.43      def
11           3.96 0.0755 61      3.80      4.11      bc
13           3.78 0.1410 61      3.50      4.06      bcd
15           3.80 0.1660 61      3.46      4.13     abcde
17           3.90 0.1700 61      3.56      4.25     abcd
19           3.92 0.0513 61      3.82      4.02      bc
21           3.97 0.1000 61      3.77      4.17     abc
23           4.12 0.1740 61      3.77      4.46     abc
25           4.15 0.1890 61      3.77      4.53     abc
27           4.34 0.0516 61      4.24      4.45      a
29           4.20 0.1270 61      3.95      4.46     ab

matrix = s:
  week_fac      emmean      SE df lower.CL upper.CL .group
0            2.56 0.1350 61      2.29      2.83      f
0.142857142857143 2.87 0.1630 61      2.55      3.20      ef
0.428571428571429 3.06 0.2720 61      2.51      3.60     bcdef
1            2.99 0.1310 61      2.72      3.25      ef
2            3.06 0.1300 61      2.80      3.32      ef
3            3.05 0.2500 61      2.55      3.55     cdef

```

4	3.26	0.2690	61	2.72	3.80	abcdef
5	3.17	0.2310	61	2.71	3.63	cdef
7	3.31	0.1420	61	3.02	3.59	cde
9	3.42	0.0592	61	3.30	3.54	de
11	3.69	0.0755	61	3.54	3.84	bcd
13	3.53	0.1410	61	3.25	3.81	bcde
15	3.93	0.1660	61	3.59	4.26	abcd
17	3.94	0.1700	61	3.60	4.28	abcd
19	3.75	0.0513	61	3.65	3.85	bc
21	3.87	0.1000	61	3.67	4.07	abc
23	3.89	0.1740	61	3.54	4.24	abcd
25	3.77	0.1890	61	3.39	4.15	abcde
27	4.12	0.0516	61	4.02	4.23	a
29	4.16	0.1270	61	3.90	4.41	ab

Degrees-of-freedom method: df.error

Confidence level used: 0.95

P value adjustment: tukey method for comparing a family of 20 estimates

significance level used: alpha = 0.05

NOTE: If two or more means share the same grouping symbol,
then we cannot show them to be different.

But we also did not show them to be the same.

ratio at weeks 0 and 29

```
em2 <- emmeans(lm2, ~ week_fac, mode = "df.error")
```

```
summary(em2) |>
```

```
dplyr::filter(week_fac == c('0', '29'))
```

week_fac	emmean	SE	df	lower.CL	upper.CL
0	2.62	0.0955	61	2.43	2.81
29	4.18	0.0896	61	4.00	4.36

Results are averaged over the levels of: matrix

Degrees-of-freedom method: df.error

Confidence level used: 0.95

Overall average at each matrix level

```
(em2 <- emmeans(lm2, ~ matrix, mode = "df.error"))
```

matrix	emmean	SE	df	lower.CL	upper.CL
ps	3.53	0.0369	61	3.45	3.60
s	3.47	0.0369	61	3.40	3.54

Results are averaged over the levels of: week_fac

Degrees-of-freedom method: df.error

Confidence level used: 0.95

```
# contrast(em2, method = "pairwise", adjust = "none", infer = TRUE)
```

```
# Remove selected objects from the environment
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
    envir = .GlobalEnv)
```

9. Convert TCID50 to PFU

based on the median and CI calculated from simulated ratios from FEV article (to be cited once the article is out). Median Log10 TCID50:PFU of -0.65 [95%CI: -1.15, -0.16]. TCID50-PFU = log ratio => PFU = TCID50 - log ratio

```
# Convert using raw data
readRDS(here('obj2', 'data', 'Obj2_dat.RDS')) |>
  dplyr::filter (microbe=='tv' & assay=='tcid50') |>
  droplevels() |>
  group_by(week) |>
  summarise(
    survival_ave = mean(survival, na.rm = TRUE)) |>
  dplyr::mutate(
    pfu_med = survival_ave + 0.65,
    pfu_lwr = survival_ave + 0.16,
    pfu_upr = survival_ave + 1.15) |>
  as.data.frame()
```

	week	survival_ave	pfu_med	pfu_lwr	pfu_upr
1	0.000	4.924	5.57	5.08	6.07
2	0.143	4.768	5.42	4.93	5.92
3	0.429	4.695	5.35	4.86	5.85
4	1.000	4.518	5.17	4.68	5.67
5	2.000	4.368	5.02	4.53	5.52
6	3.000	4.269	4.92	4.43	5.42
7	4.000	4.014	4.66	4.17	5.16
8	5.000	3.796	4.45	3.96	4.95
9	7.000	3.591	4.24	3.75	4.74
10	9.000	3.379	4.03	3.54	4.53
11	11.000	3.063	3.71	3.22	4.21
12	13.000	2.969	3.62	3.13	4.12
13	15.000	2.640	3.29	2.80	3.79
14	17.000	2.285	2.93	2.44	3.43
15	19.000	2.146	2.80	2.31	3.30
16	21.000	1.979	2.63	2.14	3.13
17	23.000	1.646	2.30	1.81	2.80
18	25.000	1.479	2.13	1.64	2.63
19	27.000	1.063	1.71	1.22	2.21
20	29.000	0.957	1.61	1.12	2.11

```
# Remove selected objects from the environment
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
    envir = .GlobalEnv)
```


10. Weather/soil conditions

10.1 Soil moisture changes

```
dat1 <- tidyr::tibble(  
  week = factor(rep(c("0", "29"), each = 6)),  
  matrix = factor(rep(c("s", "ps"), each = 3, times = 2)),  
  moisture = as.numeric(c(  
    4.319, 4.102, 5.086,  
    5.932, 5.434, 5.330,  
    4.096, 3.347, 4.096,  
    3.717, 3.635, 4.589)  
  ))
```

```
# Average moisture
```

```
dat1 |>  
  group_by(matrix, week) |>  
  do({  
    data.frame(  
      moisture_per = mean(.$moisture),  
      sd = stats::sd(.$moisture))  
  })
```

```
# A tibble: 4 × 4
```

```
# Groups:   matrix, week [4]
```

	matrix	week	moisture_per	sd
	<fct>	<fct>	<dbl>	<dbl>
1	ps	0	5.57	0.322
2	ps	29	3.98	0.529
3	s	0	4.50	0.517
4	s	29	3.85	0.432

```
dat1 |>  
  group_by(week) |>  
  do({  
    data.frame(  
      moisture_per = mean(.$moisture),  
      sd = stats::sd(.$moisture))  
  })
```

```
# A tibble: 2 × 3
```

```
# Groups:   week [2]
```

	week	moisture_per	sd
	<fct>	<dbl>	<dbl>
1	0	5.03	0.698
2	29	3.91	0.438

```
fit1 <- aov(moisture ~ matrix * week, data=dat1)
```

```
joint_tests (fit1)
```

model term	df1	df2	F.ratio	p.value
matrix	1	8	5.130	0.0532
week	1	8	17.990	0.0028
matrix:week	1	8	3.090	0.1167

```
emmeans(fit1, ~ matrix * week) |>
  pairs()
```

contrast	estimate	SE	df	t.ratio	p.value
ps week0 - s week0	1.063	0.374	8	2.850	0.0830
ps week0 - ps week29	1.585	0.374	8	4.240	0.0120
ps week0 - s week29	1.719	0.374	8	4.600	0.0080
s week0 - ps week29	0.522	0.374	8	1.400	0.5340
s week0 - s week29	0.656	0.374	8	1.760	0.3580
ps week29 - s week29	0.134	0.374	8	0.360	0.9830

P value adjustment: tukey method for comparing a family of 4 estimates

10.2 Monthly soil temperature

Data were collected from the UF/IFAS Florida Automated Weather Network (FAWN):
<https://fawn.ifas.ufl.edu> (Live Oak, FL station)

```
dat1 <- read.csv(here('obj2', 'data', 'FAWN_report_2022_2025_daily.csv')) |>
  dplyr::mutate(
    Tweather.c = (`X10m.T.avg..F.` - 32) * (5/9),
    Tsoil.c = (`Tsoil.avg.10cm...F.` - 32) * (5/9),
    rain.cm = `X2m.Rain.tot..in.` * 2.54
  )
```

```
str (dat1)
```

```
'data.frame':  1096 obs. of  18 variables:
 $ FAWN.Station      : chr  "Live Oak" "Live Oak" "Live Oak" "Live
Oak" ...
 $ Period            : chr  "1 Aug 2022" "2 Aug 2022" "3 Aug 2022"
"4 Aug 2022" ...
 $ X10m.T.avg..F.    : num  81.6 82.2 80.7 80.8 82.4 ...
 $ X10m.T.min..F.    : num  72.9 72.4 73.6 71.2 73.6 ...
 $ X10m.T.max..F.    : num  93.1 93.4 92.6 92.1 94.2 ...
 $ Tsoil.avg.10cm...F. : num  93.7 92.9 94.3 91.6 93.2 ...
 $ Tsoil.min.avg..10cm...F. : num  88.4 86.4 88.9 84.7 86.3 ...
 $ Tsoil.max.avg..10cm...F. : num  99.5 100.4 101.1 99.7 100.8 ...
 $ X2m.DewPt.avg..F. : num  75.4 74.3 74.3 74.8 74.9 ...
 $ RelHum.avg.2m...pct. : int  82 77 81 81 78 87 81 80 83 84 ...
 $ X2m.Rain.tot..in. : num  0.01 0 0.2 0 0 0 0 0 0.02 0 ...
 $ X2m.Rain.max.over.15min..in.: num  0.01 0 0.1 0 0 0 0 0 0.02 0 ...
 $ SolRad.avg2m...w.m.2. : num  205 269 260 260 258 ...
 $ BP.avg..mb.       : int  1016 1013 1012 1013 1014 1016 1016 1015
1015 1015 ...
 $ N....obs.         : int  96 96 96 96 96 96 96 96 96 96 ...
```

```

$ Tweather.c          : num  27.6 27.9 27.1 27.1 28 ...
$ Tsoil.c             : num  34.3 33.9 34.6 33.1 34 ...
$ rain.cm             : num  0.0254 0 0.508 0 0 0 0 0 0.0508 0 ...

dat1_monthly <- dat1 |>
  mutate(
    Period_clean = str_squish(Period),
    date = dmy(Period_clean),
    month = month(date, label = TRUE, abbr = TRUE) # Jan, Feb, Mar ...
  ) |>
  group_by(month) |>
  summarise(
    Tweather.ave.c = round(mean(Tweather.c, na.rm = TRUE), 2),
    Tweather.sd.c = round(sd(Tweather.c, na.rm = TRUE), 2),
    Tsoil.ave.c = round(mean(Tsoil.c, na.rm = TRUE), 2),
    Tsoil.sd.c = round(sd(Tsoil.c, na.rm = TRUE), 2),
    RH.ave = round(mean(RelHum.avg.2m...pct., na.rm = TRUE), 2),
    RH.sd = round(sd(RelHum.avg.2m...pct., na.rm = TRUE), 2),
    rain.ave = round(mean(rain.cm, na.rm = TRUE), 3),
    rain.sd = round(sd(rain.cm, na.rm = TRUE), 3),
    SolarRad.ave = round(mean(SolRad.avg2m...w.m.2., na.rm = TRUE), 2),
    SolarRad.sd = round(sd(SolRad.avg2m...w.m.2., na.rm = TRUE), 2),
    .groups = "drop"
  ) |>
  arrange(match(month, month.abb))

dat1_monthly

# A tibble: 12 × 11
  month Tweather.ave.c Tweather.sd.c Tsoil.ave.c Tsoil.sd.c RH.ave RH.sd
  <ord>      <dbl>      <dbl>      <dbl>      <dbl> <dbl> <dbl>
1 Jan         11.7         5.35         12.7         3.5   76.0 10.9
2 Feb         15.7         4.02         16.7         3.24  77.5 10.2
3 Mar         17.9         3.48         19.9         2.7   72.7 12.5
4 Apr         20.9         2.84         24.2         2.47  72.0  9.59
5 May         24.0         1.92         27.9         2.52  76.8  8.22
6 Jun         26.3         1.79         30.2         2.64  79.2  7.5
7 Jul         27.4         1.26         31.1         1.68  83.1  5.48
8 Aug         27.1         1.4          31.3         2.03  82.7  5.45
9 Sep         25.0         1.49         28.7         1.85  82.0  8.06
10 Oct         20.5         3.11         24.4         2.17  75.7  8.93
11 Nov         17.7         4.58         20.3         3.8   80.5 10.2
12 Dec         13.4         5.06         14.9         3.4   79.1 11.2
# i 4 more variables: rain.ave <dbl>, rain.sd <dbl>, SolarRad.ave <dbl>,
#   SolarRad.sd <dbl>

# Warm and cool seasons
dat1 |>
  mutate(
    Period_clean = str_squish(Period),

```

```

date = dmy(Period_clean),
month_num = month(date),
season = case_when(
  month_num %in% c(10, 11, 12, 1, 2, 3) ~ "Cool season (Oct-Mar)",
  month_num %in% c(4:9) ~ "Warm season (Apr-Sep)") |>
group_by(season) |>
summarise(
  Tsoil.ave.c = mean(Tsoil.c, na.rm = TRUE),
  Tsoil.sd.c = sd(Tsoil.c, na.rm = TRUE),
  .groups = "drop")

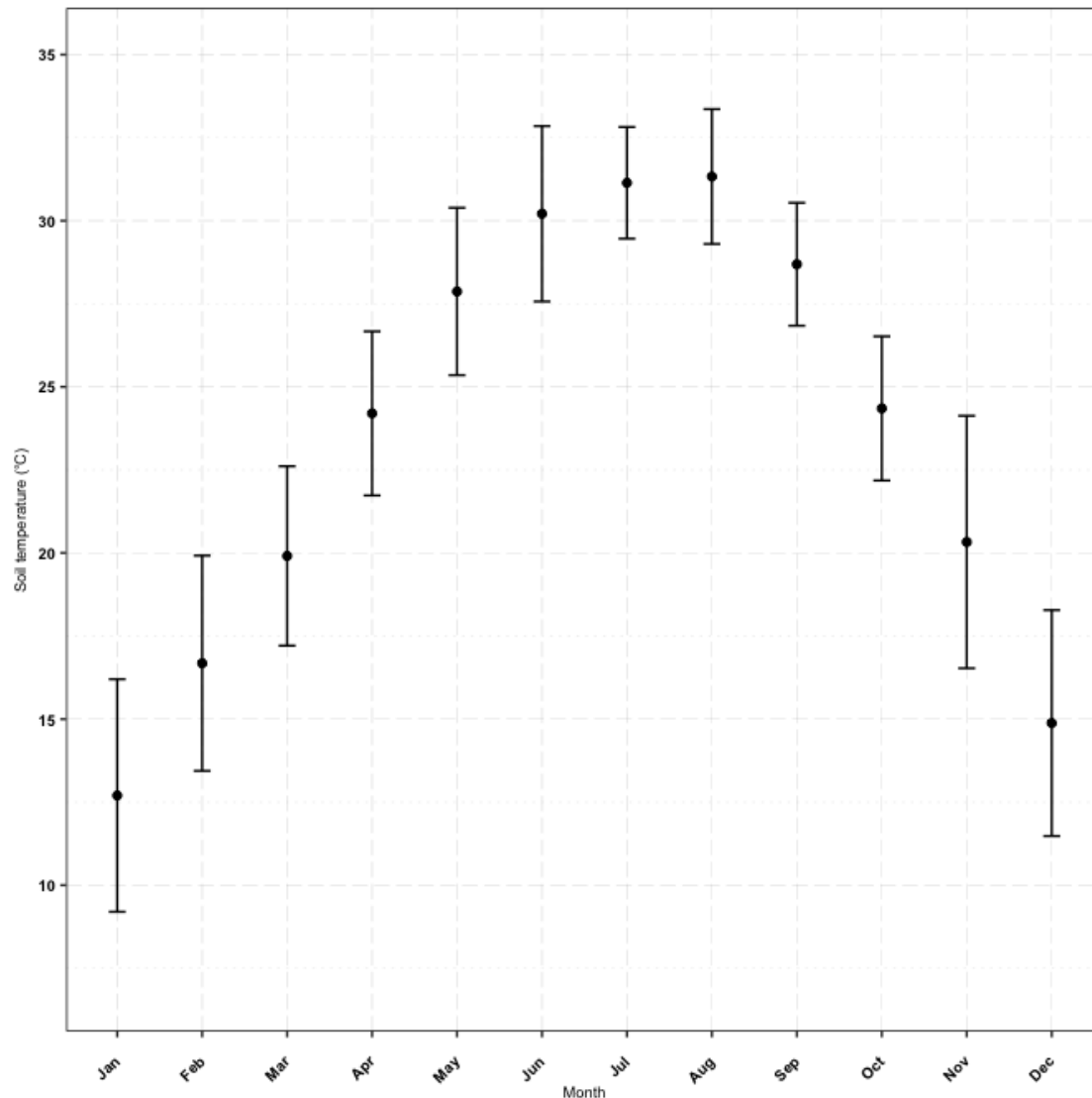
# A tibble: 2 × 3
  season          Tsoil.ave.c Tsoil.sd.c
  <chr>          <dbl>      <dbl>
1 Cool season (Oct-Mar)    18.2      5.01
2 Warm season (Apr-Sep)   28.9      3.29

# Plot monthly averages
p_m2 <- ggplot(dat1_monthly,
  aes(x = month, y = Tsoil.ave.c)) +
  geom_point(size = 1.5, color = "black") +
  geom_errorbar(aes(ymin = Tsoil.ave.c - Tsoil.sd.c,
    ymax = Tsoil.ave.c + Tsoil.sd.c),
    width = 0.2, color = "black") +
  labs(
    x = "Month",
    y = "Soil temperature (°C)") +
  scale_y_continuous(
    breaks = seq(10, 35, 5),
    labels = seq(10, 35, 5),
    limits = c(7, 35)) +
  theme_bw() +
  theme(text = element_text(family = "Arial"),
    legend.position = 'inside', legend.position.inside = c(0.86, 0.18),
    legend.direction = "vertical",
    legend.text = element_text(size = 7),
    legend.title = element_text(size = 7, face = "bold"),
    legend.key.height = unit(0.4, "cm"),
    axis.text.x = element_text(angle = 45, size = 7, color = "black",
face = "bold", hjust = 1, vjust = 0.5),
    axis.text.y = element_text(size = 7, color = "black", face = "bold"),
    axis.title.x = element_text(size = 7, vjust = -1),
    axis.title.y = element_text(size = 7, vjust = 2),
    panel.border = element_rect(color = "black", fill = NA, linewidth =
0.5),
    panel.grid.major = element_line(color = "gray", linewidth = 0.1,
linetype = "longdash"),
    panel.grid.minor = element_line(color = "lightgray", linewidth = 0.1,
linetype = "dotted"),
    panel.background = element_rect(fill = "transparent"),

```

```
plot.background = element_rect(fill = "transparent", color =
"transparent"))
```

p_m2



```
ggsave(file = here('obj2', 'results', 'Fig S1 ave_monthly_soil_temp.tiff'),
plot = p_m2,
width = 7, height = 5, units = c("cm"), dpi=400)
```

```
rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
envir = .GlobalEnv)
```

10.3 Seven days prior to sample collection

```
dat1 <- read.csv(here('obj2', 'data', 'FAWN_report_seven_day_hourly.csv')) |>
dplyr::mutate(
  Tweather.c = (`X10m.T.avg..F.`-32) * (5/9),
```

```

    Tsoil.c = (`Tsoil.avg.10cm...F.` -32) * (5/9),
    rain.cm = `X2m.Rain.tot..in.` * 2.54
  )

str (dat1)

'data.frame':   192 obs. of  18 variables:
 $ FAWN.Station      : chr  "Live Oak" "Live Oak" "Live Oak" "Live
Oak" ...
 $ Period            : chr  "22 Mar 2024 12:00 AM" "22 Mar 2024
1:00 AM" "22 Mar 2024 2:00 AM" "22 Mar 2024 3:00 AM" ...
 $ X10m.T.avg..F.    : num  64.8 65 64.9 62.4 61.3 ...
 $ X10m.T.min..F.    : num  64.7 64.8 64.3 61.1 61.2 ...
 $ X10m.T.max..F.    : num  65 65.2 65.2 63.8 61.6 ...
 $ Tsoil.avg.10cm...F. : num  70.2 69.6 69.1 68.7 68.4 ...
 $ Tsoil.min.avg..10cm...F. : num  69.9 69.4 68.9 68.6 68.2 ...
 $ Tsoil.max.avg..10cm...F. : num  70.4 69.8 69.3 68.8 68.5 ...
 $ X2m.DewPt.avg..F.  : num  54 55.9 57 58.5 59.7 ...
 $ RelHum.avg.2m...pct. : int  69 73 77 88 94 96 97 96 96 96 ...
 $ X2m.Rain.tot..in.  : num  0 0 0 0.02 0.05 0.25 0.37 0.03 0.04
0.08 ...
 $ X2m.Rain.max.over.15min..in.: num  0 0 0 0.01 0.03 0.15 0.19 0.02 0.02
0.03 ...
 $ SolRad.avg2m...w.m.2. : num  0 0 0 0 0 ...
 $ BP.avg..mb.        : int  1012 1013 1012 1011 1010 1010 1010 1010
1010 1010 ...
 $ N....obs.          : int  4 4 4 4 4 4 4 4 4 4 ...
 $ Tweather.c         : num  18.2 18.3 18.3 16.9 16.3 ...
 $ Tsoil.c            : num  21.2 20.9 20.6 20.4 20.2 ...
 $ rain.cm            : num  0 0 0 0.0508 0.127 ...

dat1 |>
  mutate(
    Period_clean = str_squish(Period),
    datetime = parse_date_time(Period_clean, orders = "d b Y I:M p"),
    date = as_date(datetime)
  ) |>
  group_by(date) |>
  summarise(
    Tweather.ave.c = round(mean(Tweather.c, na.rm = TRUE), 3),
    Tweather.sd.c = round(sd(Tweather.c, na.rm = TRUE), 3),
    Tsoil.ave.c = round(mean(Tsoil.c, na.rm = TRUE), 3),
    Tsoil.sd.c = round(sd(Tsoil.c, na.rm = TRUE), 3),
    RH.ave = round(mean(RelHum.avg.2m...pct., na.rm = TRUE), 3),
    RH.sd = round(sd(RelHum.avg.2m...pct., na.rm = TRUE), 3),
    rain.ave = round(mean(rain.cm, na.rm = TRUE), 3),
    rain.sd = round(sd(rain.cm, na.rm = TRUE), 3),
    SolarRad.ave = round(mean(SolRad.avg2m...w.m.2., na.rm = TRUE), 3),
    SolarRad.sd = round(sd(SolRad.avg2m...w.m.2., na.rm = TRUE), 3),

```

```

    .groups = "drop"
  )
# A tibble: 8 × 11
  date       Tweather.ave.c Tweather.sd.c Tsoil.ave.c Tsoil.sd.c RH.ave RH.sd
<date>         <dbl>         <dbl>         <dbl>         <dbl> <dbl> <dbl>
1 2024-03-22         18.2           0.965          19.6           0.622  91.3  7.50
2 2024-03-23         19.2           1.71           20.1           1.59   84.9  11.2
3 2024-03-24         15.8           2.76           19.0           1.60   75.2  11.3
4 2024-03-25         16.7           4.42           18.2           2.1    77.5  12.0
5 2024-03-26         21.3           4.04           20.0           1.98   74.3  12.1
6 2024-03-27         21.1           1.01           21.1           0.423  87.8  5.34
7 2024-03-28         18.6           2.68           20.4           1.35   76.0  20.7
8 2024-03-29         16.0           3.94           18.6           2.17   51.4  18.6
# i 4 more variables: rain.ave <dbl>, rain.sd <dbl>, SolarRad.ave <dbl>,
#   SolarRad.sd <dbl>

# Average weekly weather conditions prior to sample collection
dat1 |>
  summarise(Tweather.ave.c = mean(Tweather.c, na.rm = TRUE),
            Tweather.sd.c = sd(Tweather.c, na.rm = TRUE),
            Tsoil.avg.c = mean(Tsoil.c, na.rm = TRUE),
            Tsoil.sd.c = sd(Tsoil.c, na.rm = TRUE),
            RH.avg = mean(`RelHum.avg.2m...pct.` , na.rm = TRUE),
            RH.sd = sd(`RelHum.avg.2m...pct.` , na.rm = TRUE),
            rain.avg.cm = mean((`X2m.Rain.tot..in.`*2.54), na.rm = TRUE),
            rain.sd.cm = sd((`X2m.Rain.tot..in.`*2.54), na.rm = TRUE),
            SolaRad.avg = mean(`SolRad.avg2m...w.m.2.` , na.rm = TRUE),
            SolaRad.sd = sd(`SolRad.avg2m...w.m.2.` , na.rm = TRUE),
            .groups = "drop") |>
  as.data.frame() |>
  mutate(across(where(is.numeric), ~ round(.x, 2)))

  Tweather.ave.c Tweather.sd.c Tsoil.avg.c Tsoil.sd.c RH.avg RH.sd
rain.avg.cm
1          18.3          3.54          19.6          1.82   77.3  17.4
0.05
  rain.sd.cm SolaRad.avg SolaRad.sd
1          0.21         168         250

rm(list = setdiff(ls(envir = .GlobalEnv, all.names = TRUE), list_exc),
  envir = .GlobalEnv)

```