pQMRA of ingested water in an IWS - Cajibio, Colombia

Angela Bayona-Valderrama

This is the participatory modelling approach (PM) to estimating the potential health risks of ingesting fecally contaminated water in an intermittent supply system (IWS), the data comes from grab water samples taken in 2022 and from self-reported daily volumes of water ingestion obtained in 2023- the project was carried out in an IWS system in Cajibio, Colombia.

Load packages

```
library(tidyverse)
library(dplyr)
library(here)
library(rriskDistributions)
library(VGAM)
library(ggridges)
library(scales)
library(hrbrthemes)
library(rstatix)
library(kableExtra)
library(flextable)
library(fitdistrplus)
library(univariateML)
library(EnvStats)
```

Load data

Loading collected data, coming from two consecutive phases

The first holds results from measuring Thermotolerant coliform (TTC) concentrations in drinking water in a cross-sectional study from 200 households in 2022. It includes water quality measurements of paired grab water samples taken at the tap and at the point of storage, and responses to household survey.

The second holds results from implementing "Risk Dialogues" workshop with community leaders in 2023. It includes self-reports of daily volumes of water ingestion (ml), drinking water habits, sources of drinking water arriving to the household, and household drinking water management.

```
data <- read.csv(here("data","clean_df.csv"))

data_2023 <- read.csv(here("data", "RiskDialogues_raw.csv"))
```

Probability distribution fitting for volumes of ingested water

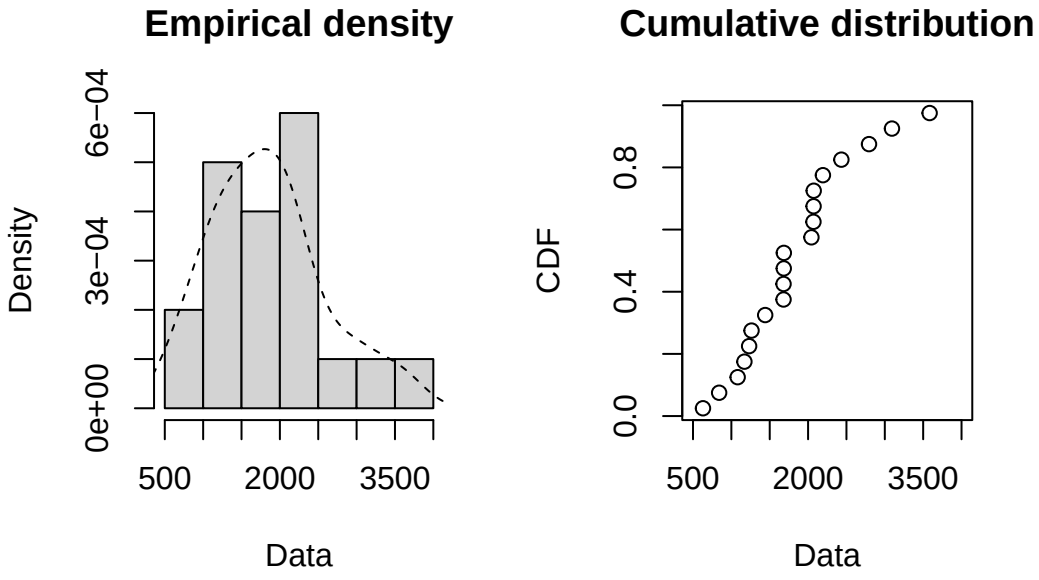
Below, we follow Delignette-Muller (2023) to fit a PDF to the self-reported volume of ingestion of water data.

```
#brief check of the Vol_tot data
summary(data_2023$Vol_tot)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    630    1255    1684    1837    2104    3584
```

First, we visually inspect the data by plotting a density histogram and a CDF

```
#Visual examination of data in a density histogram and a CDF

plotdist <- plotdist(data_2023$Vol_tot, histo = TRUE, demp = TRUE)
```

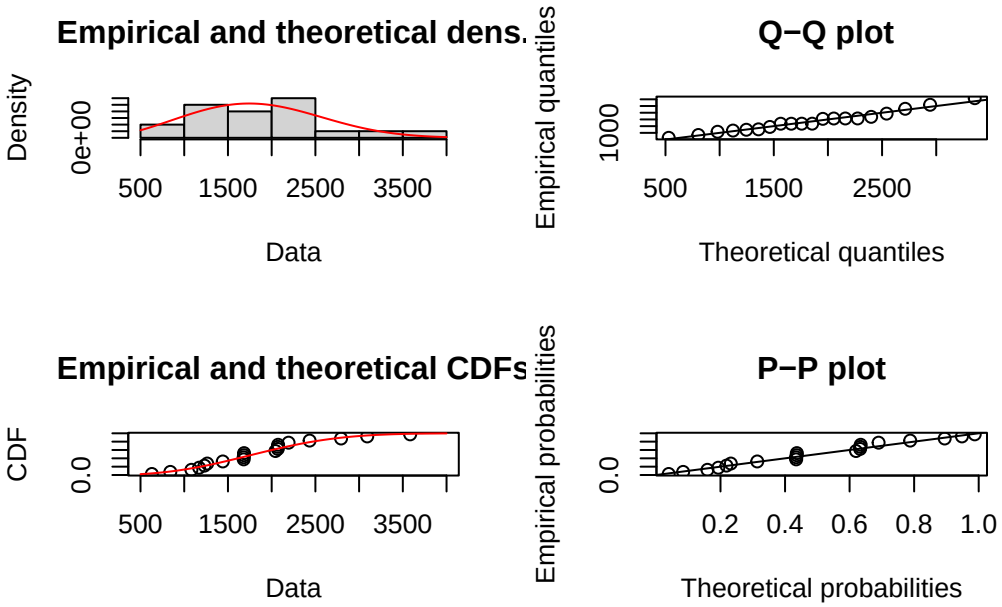


It is not possible to clearly define a probability function from the visual inspection. So, we fit the PDF candidates using MLE (for full explanation of MLE see the aQMRA code from Chapter 1 in my PhD thesis), and assess the Akaike Information Criterion (AIC) for each one.

```
#Fitting a weibull distribution
fit.weibull <- fitdist(data_2023$Vol_tot, "weibull",method = "mle")
fit.weibull$aic
```

```
[1] 323.4365
```

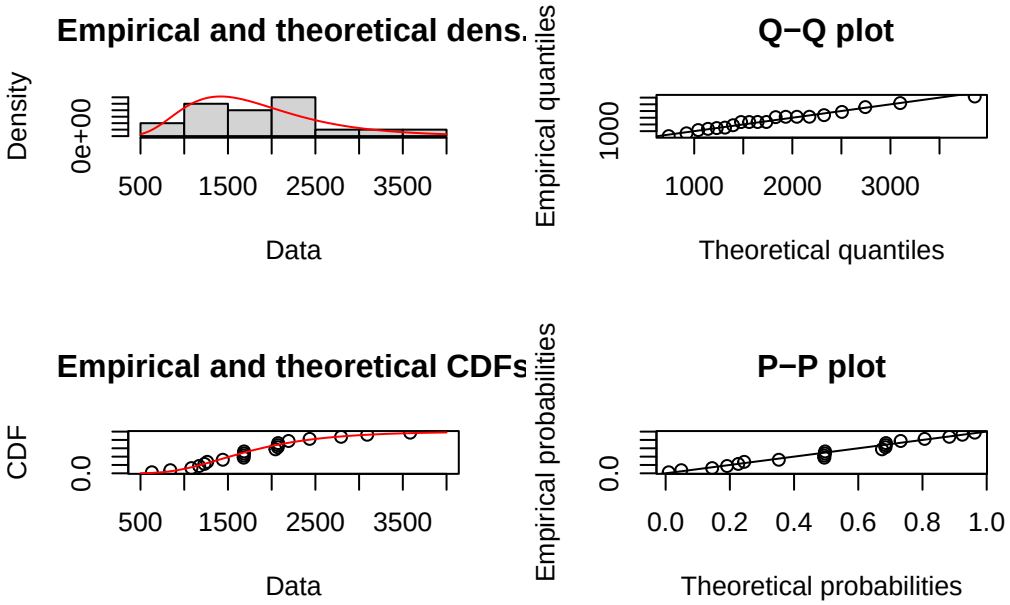
```
plot(fit.weibull)
```




```
#fitting a lognormal distribution
fit.lognormal <- fitdist(data_2023$Vol_tot, "lnorm", method = "mle")
fit.lognormal$aic
```

```
[1] 323.4822
```

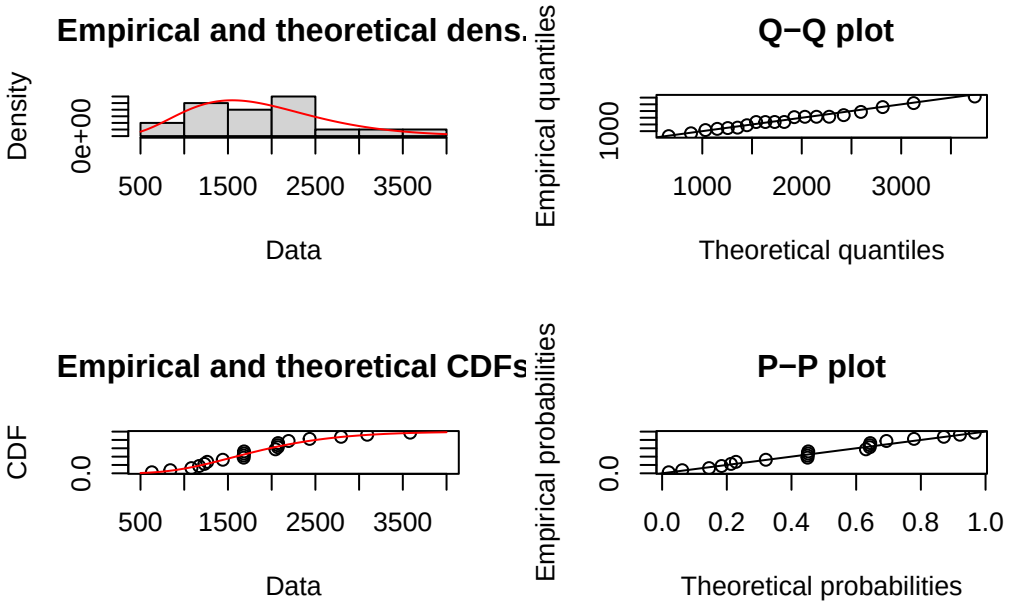
```
plot(fit.lognormal)
```




```
#fitting a gamma distribution
fit.gamma <- fitdist(data_2023$Vol_tot, "gamma", method = "mle")
fit.gamma$aic
```

```
[1] 322.9684
```

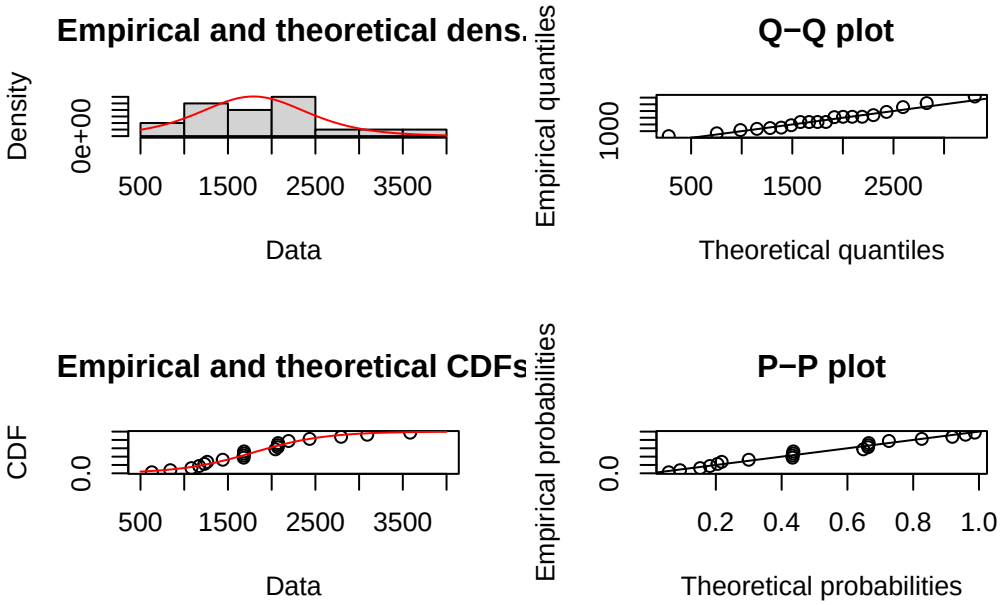
```
plot(fit.gamma)
```




```
#fitting a beta distribution
fit.logis <- fitdist(data_2023$Vol_tot, "logis", method = "mle")
fit.logis$aic
```

```
[1] 324.6645
```

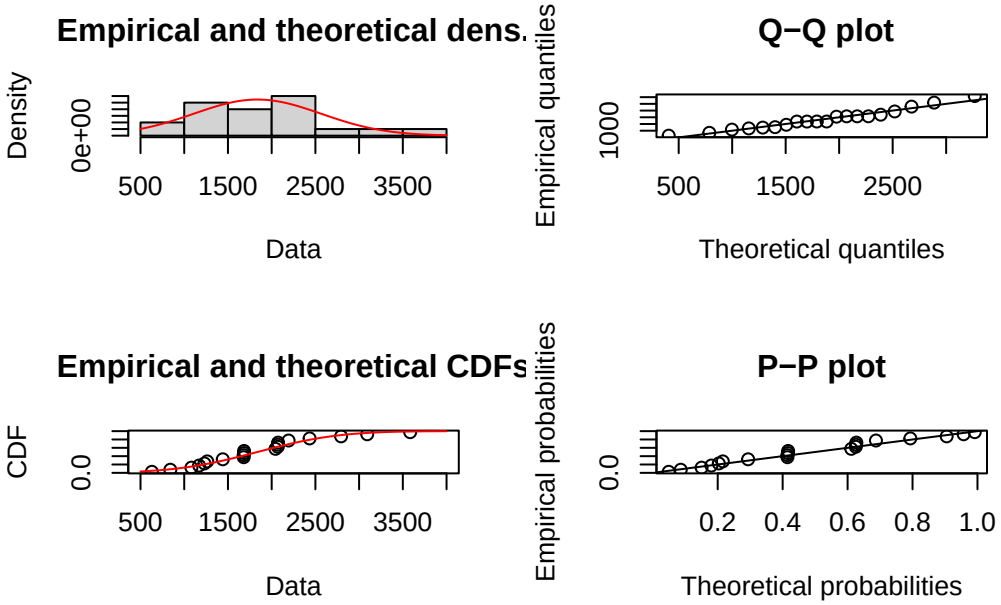
```
plot(fit.logis)
```



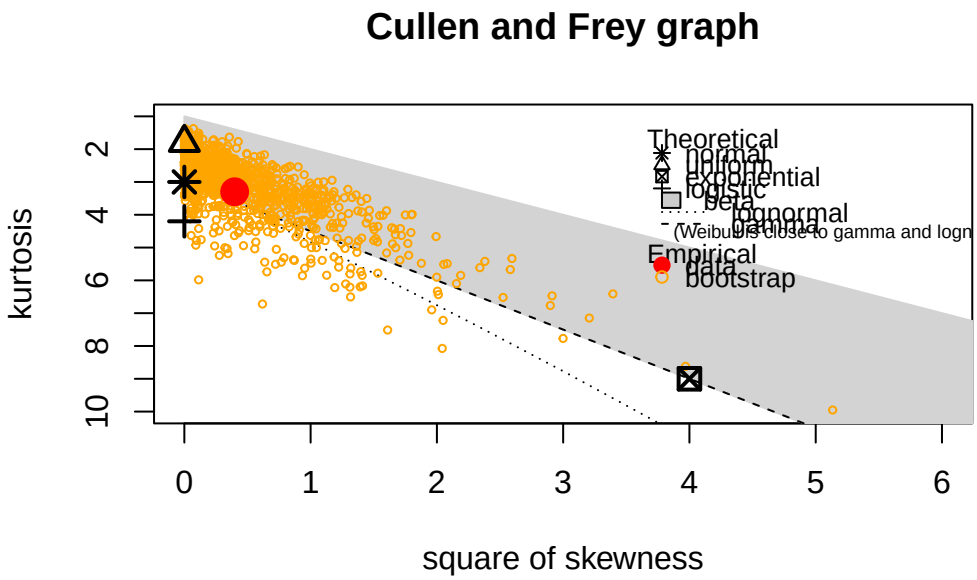

```
fit.normal <- fitdist(data_2023$Vol_tot, "norm", method = "mle")
fit.normal$aic
```

```
[1] 324.4641
```

```
plot(fit.normal)
```

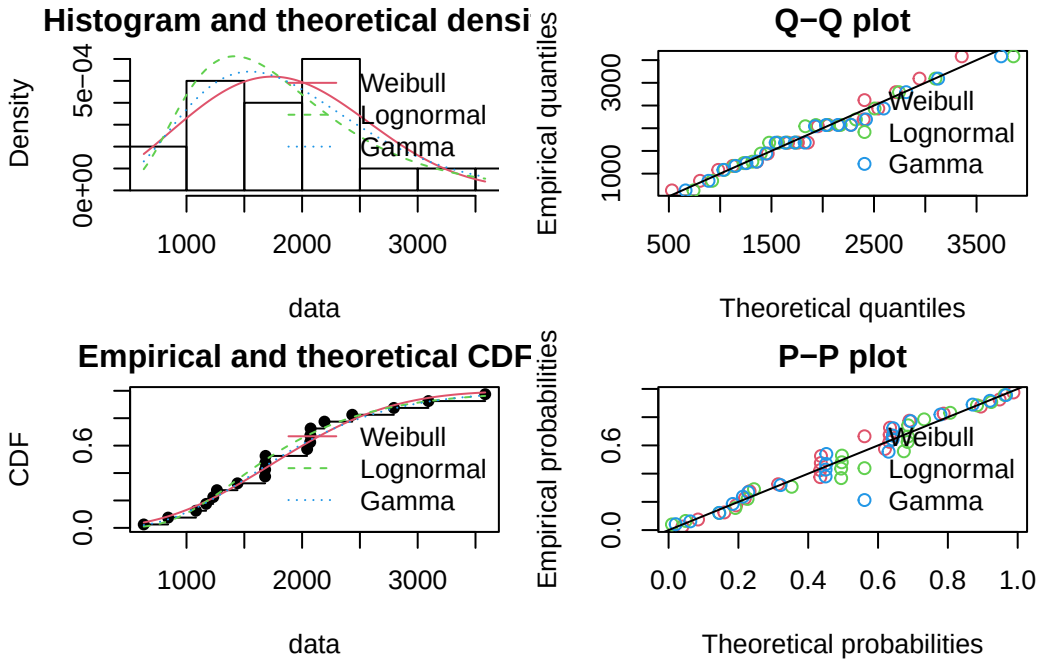


```r
#Fit of distributions by MLE using fitdist
descdist(data_2023$Vol_tot, boot = 1000)
```




```
summary statistics
------
min:  630   max:  3584
median:  1683.5
mean:  1837.125
estimated sd:  748.6786
estimated skewness:  0.6312181
estimated kurtosis:  3.299034
```

```r
#Goodness of fit plots
par(mfrow=c(2,2), mar=c(4, 4, 2, 1))
plot.legend <- c("Weibull", "Lognormal", "Gamma")
denscomp(list(fit.weibull, fit.lognormal, fit.gamma), legendtext = plot.legend)
qqcomp(list(fit.weibull, fit.lognormal, fit.gamma), legendtext = plot.legend)
cdfcomp(list(fit.weibull, fit.lognormal, fit.gamma), legendtext = plot.legend)
ppcomp(list(fit.weibull, fit.lognormal, fit.gamma), legendtext = plot.legend)
```



The AIC and visual inspection indicate the gamma PDF could be the best one. Through the code below we obtain the parameter estimates for the fitted Gamma distribution.

```
#estimated parameters for gamma distribution
fit.gamma$estimate
```

```
      shape        rate
5.602465935 0.002969927
```

Double-checking by using univariateML package to assess multiple densities

```
AIC(
  mlinvgamma(data_2023$Vol_tot),
  mlgamma(data_2023$Vol_tot),
  mllnorm(data_2023$Vol_tot),
  mlweibull(data_2023$Vol_tot),
  mlinvweibull(data_2023$Vol_tot)
)
```

```
                                df      AIC
mlinvgamma(data_2023$Vol_tot)    2 325.1706
mlgamma(data_2023$Vol_tot)       2 322.8008
mllnorm(data_2023$Vol_tot)       2 323.4822
mlweibull(data_2023$Vol_tot)     2 323.4365
mlinvweibull(data_2023$Vol_tot)  2 328.6280
```

Through the code below we double-check the parameter estimates for the fitted Gamma distribution.

```
mlgamma(data_2023$Vol_tot)
```

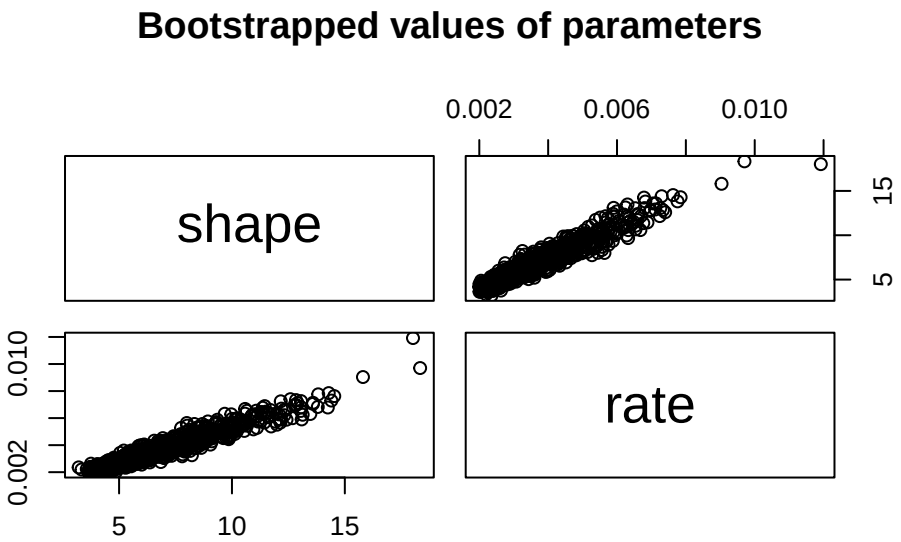
```
Maximum likelihood estimates for the Gamma model
   shape      rate
6.152859  0.003349
```

Finally, we assess uncertainty in parameter estimates. Parametric and non-parametric bootstraps can be used to assess uncertainty in the parameters of the fitted gamma distribution.

```
#Uncertainty in parameter estimates using bootdist
gamma.B <- bootdist(fit.gamma, niter = 1000)
summary(gamma.B)
```

```
Parametric bootstrap medians and 95% percentile CI
           Median        2.5%        97.5%
shape 6.227352995 3.949001667 12.391692686
rate  0.003335316 0.002077936  0.006603521

The estimation method converged only for 953 among 1000 iterations
```

```
#Uncertainty in parameter estimates using bootdist, plot
plot(gamma.B)
```



Having chosen the gamma distribution for volume (ml) of ingested data, we follow the same code as aQMRA to assess infection and illness risks.

Maximum likelihood estimation

Vector for analysis of TTCs in stored water

Saving stored TTCs concentration data for MLE in two separate vectors. NA values are dropped. Zero values will be assumed as < 1 in the MLE.

```
#vector for tap water data
T_TTC <- data %>%
  drop_na(T_TTC) %>%
  pull(T_TTC)
```

```
#vector for stored water data
S_TTC <- data %>%
  drop_na(S_TTC) %>%
  pull(S_TTC)
```

Probability density function fitting

To perform a MLE in R one defines a negative log likelihood function and minimizes it, since R's standard optimisation, or gradient search, follows a minimisation routine.

The following code defines a negative log likelihood function for the TTC concentrations found in stored water. Note that zero values are subset to calculate the probability between 0 and 1 (sum0), while the TNTC values are subset and treated differently by calculating the probability of obtaining a value greater than 551 CFU (sum2).

```
negloglike = function(parameters,
                      threshold,
                      counts){

  mean = parameters[1]
  sd = parameters[2]

  counts0 = counts[counts < 1]
  sum0 = sum(-log(pnorm(log(counts0+1), mean=mean, sd = sd)))

  counts1 = counts[counts >= 1 & counts<=threshold]
  sum1 = sum(-log(dnorm(log(counts1), mean=mean, sd = sd)))

  # Probabilities and likelihoods for the TNC values (>551)
```

```
  counts2 = counts[counts>threshold]

  # Setting a flag for the counts using the threshold
  prob2 = 1 - pnorm(log(counts2), mean = mean, sd = sd)
  sum2 = sum(-log(prob2))

  return(sum0+sum1+sum2)
}
```

This is a test of the `negloglike` function for the TTC data of stored water.

```
negloglike(parameters = c(mean(S_TTC),sd(S_TTC)),
           threshold = 551,
           counts =  2)
```

```
[1] 7.01512
```

This is a test of the `negloglike` function for the TTC data of tap water

```
negloglike(parameters=c(mean(T_TTC),sd(T_TTC)),
           threshold = 551,
           counts=2)
```

```
[1] 4.865639
```

To identify the MLE estimates, the negloglike function is minimised using the parameters that define the lognormal distribution and the `nlm` function. The following code defines this parameters for the concentration of stored TTCs

```
out_stored <- nlm(negloglike,
           p = c(mean(S_TTC,
                      na.rm = TRUE), sd(S_TTC, na.rm = TRUE)),
           hessian=TRUE,
           threshold = 551,
           counts = S_TTC)

out_stored
```

```
$minimum
[1] 232.1302

$estimate
[1] 4.999007 7.309606
```

```
$gradient
[1] -3.932640e-05 -1.675455e-05

$hessian
           [,1]       [,2]
[1,]  2.0841030 -0.3804906
[2,] -0.3804906  1.1862280

$code
[1] 1

$iterations
[1] 25
```

As for the tap water concentrations, the log transformed data will be fitted to a normal distribution. The same `negloglike` function can be used for this set of data.

```
#| message: false

out_tap <- nlm(negloglike,
           p = c(mean(T_TTC, na.rm = TRUE),
                 sd(T_TTC, na.rm = TRUE)),
           hessian=TRUE,
           threshold = 551,
           counts = T_TTC)

out_tap
```

```
$minimum
[1] 82.16034

$estimate
[1] -6.906716  4.974429

$gradient
[1] 3.508108e-06 6.576310e-06

$hessian
         [,1]     [,2]
[1,] 2.466257 3.681995
[2,] 3.681995 6.295158

$code
[1] 1
```

```
$iterations
[1] 22
```

The following code uses the outcome of the previous MLE, point and range estimates taken from literature, and Montecarlo tecniques to assess acute diahrreal disease (ADD) risk of infection and illness given the ingestion of microbiologically contaminated drinking water in an intermittent water supply system. The code is organised to follow the typical QMRA framework of Exposure Assessment, Dose-Response calculation, and Risk Characterisation (Haas et al 1999, Haas et al 2014).

Exposure assessment

First, a random number generator seed and a predefined number of iterations are set

```
#set random number generator seed
set.seed(123)
iter=10^4
```

Second, point estimates for risk calculations (from literature) are defined

```
#####Input of point estimates
#pathogen info
morbidity <- 200/1000 # (80 to 200/1000inhab) national prevalence of ADD in children under
```

Pathogen concentrations

Pathogen concentrations are estimated using the results from the previous MLE and point estimates from literature (ETEC=Enterotoxigenic E Coli, Campy=Campylobacter jejuni, rota=rotavirus)

Estimation of TTC concentrations from MLE

```
#####Input of ranges
    ####Ranges of concentration of TTCs in tap and stored water are evaluated by
    ####sampling data from the distributions estimated in the MLE section

#concentrations in tap water from distribution fitted in MLE in previous section
tap = exp(rnorm(iter, mean = out_tap$estimate[1],sd = out_tap$estimate[2])) #conc of TTCs

#concentrations in stored water from distribution fitted in MLE in previous section
stored = exp(rnorm(iter, mean = out_stored$estimate[1],
                   sd= out_stored$estimate[2]))#conc of TTCs in stored water (CFU/100ml)
```

Pathogen ratios to estimate concentration

In the code below the concentration of pathogens is estimated using the TTC estimates from the MLE, the point estimates and PDFs taken from literature

```
sim_conc <- rbind(tibble(w_source = "tap", conc_0 = tap),
      tibble(w_source = "stored", conc_0 = stored))

sim_conc %>%
  summarise(count = n(),
            min.conc=min(conc_0),
            mean.conc = mean(conc_0),
            median.conc = mean(conc_0),
            p95.conc = quantile(conc_0,0.95),
            p5.conc = quantile(conc_0,0.05),
            max.conc=max(conc_0),
            .by = w_source)
```

```
# A tibble: 2 x 8
  w_source count min.conc     mean.conc   median.conc  p95.conc p5.conc max.conc
  <chr>    <int>    <dbl>         <dbl>         <dbl>     <dbl>   <dbl>    <dbl>
1 tap      10000 4.93e-12          54.3          54.3    3.53e0 2.84e-7  2.06e 5
2 stored   10000 1.25e- 9 23565004341.  23565004341.     2.48e7 8.76e-4  1.27e14
```

```
sim_conc_pathogens <- sim_conc %>%
  mutate(
    #estimation of concentration of ETEC from tap and stored TTC MLE
    ETEC = (conc_0 * 0.076), #from Barragan et al 2021

    #estimation of concentration of Campy from tap and stored TTC MLE
    campy = conc_0/(1+1/rlnorm(nrow(sim_conc), 0.0089, 1.33)), #from Bivins2017 and adapte

    #campy = conc_0 * 0.95 * rlnorm(nrow(sim_conc), 0.0089, 1.33), #from Bivins2017, direc

    #estimation of concentration of rota from tap and stored TTC MLE
    #rota = conc_0 * rlnorm(nrow(sim_conc), 8.79e-7, 1.77e-6)) |> #from Bivins2017
    rota=conc_0/(1+1/rlnorm(nrow(sim_conc), 8.79e-7, 1.77e-6)))%>%

  # Converting the table to long format
pivot_longer(cols = ETEC:rota, names_to = "pathogen",values_to = "conc_p")

#summary table
sim_conc_pathogens %>%
  summarise(count = n(),
            min.conc_p = min(conc_p),
```

```
            mean.conc_p = mean(conc_p),
            median.conc_p = mean(conc_p),
            p95.conc_p = quantile(conc_p,0.95),
            p5.conc_p = quantile(conc_p,0.05),
            max.con_p= max(conc_p),
            .by = w_source)
```

```
# A tibble: 2 x 8
  w_source count min.conc_p  mean.conc_p median.conc_p p95.conc_p    p5.conc_p
  <chr>    <int>      <dbl>        <dbl>         <dbl>      <dbl>        <dbl>
1 tap      30000   3.75e-13         20.3          20.3       1.00 0.0000000578
2 stored   30000   9.48e-11 6933517435.   6933517435.  7041189.   0.000183
# i 1 more variable: max.con_p <dbl>
```

Volume of ingested water

Volume of ingested water is estimated using the gamma distribution defined previously. EPA (2019) and WHO (2017) canonical ingestion values are added for later comparisons (see *aQMRA_all.qmd* and *EPA_2019.R* codes for full explanation of ingestion volumes of water estimations).

Age segregated estimations of ingestion of water, EPA (2019)

The code below is implemented to fit log-normal distributions to water ingestion data (from EPA exposure Handbook 2019). The reported volumes of ingested water are organised per percentiles in individual vectors to then obtain means and standard deviations, per age group, from the fitted distribution.

Through the code below we obtain estimated parameters for the chosen PDFs (per age group), these parameters will later on be used to sample random numbers for each age group.

```
# #EPA Handbook 2019
# #Table 3-17 Two day average consumer only estimates of combined direct and
# #indirect water ingestion.
# #The code below fits truncated normal and lognormal distributions (previously #tested fo
#
# #creating a vector with percentiles
# p=c(0.01,0.05,0.10,0.25,0.5,0.75,0.90,0.95,0.99)
#
# #truncated normal pdf to estimate ingestion of water age 16 to 70 years
# vol_fitted_tnorm_70=get.tnorm.par(p = p,
#                   q = c(15,   103,    205,    503,    1024,   1784,   2645,   3250,   47
#
```

```
# #truncated normal pdf to estimate ingestion of water all ages
# vol_fitted_tnorm_all=get.tnorm.par(p = p,
#                   q = c(13,   70, 147,    369,    834,    1540,   2413,   2972,   4463))
```

Through the code below we obtain estimated parameters for the chosen PDFs (per age group), these parameters will later on be used to to sample random numbers for intake factors by age group.

```
# #EPA Handbook 2019 #Table 3-31 Total tap water intake
# #Table 3-32 General Dietary sources of tap water
# #The code below fits truncated normal distributions to the data reported in
# #Table 3-31. The resulting values are then multplied by an estimated percentage  # #fact
# #Table 3-32.


## 20 to 64 years of age, equivalent to 16_to_70 #

#Intake_fitted_70 = get.tnorm.par(p = p, q = c(12, 27, 35, 49, 61, 72, 79, 83, 90)*.92)/10

## All ages, using 20 to 64 since the table does not have this category # #Intake_fitted_a
```

Two dataframes (`IntakeFactor` and `vol_fitted_EPA_tnorm`) are created to store fitted means, standard deviations, upper and low values of truncated normal distributions (per age group).

```
# IntakeFactor <- tibble(
#   age =c("16_to_70",
#          "All_ages"),
#   bind_rows(Intake_fitted_70,
#             Intake_fitted_all)) %>%
#   rename_with(.cols = c(mean:upper),.fn = \(x) paste0(x,".fac")) %>%
#   mutate(lower.fac = if_else(lower.fac<0,0,lower.fac))
#
# vol_fitted_EPA_tnorm <- tibble(
#   age = c("16_to_70",
#           "All_ages"),
#   bind_rows(vol_fitted_tnorm_70,
#             vol_fitted_tnorm_all)) %>%
#   rename_with(.cols = c(mean:upper),.fn = \(x) paste0(x,".vol")) %>%
#   left_join( IntakeFactor,
#   by = "age")
```

Through the code below, the volume of water intake values are sampled from distributions defined by the parameters in `vol_fitted_EPA_tnorm` by age group

```
# #Volume of consumed water in litres
# sim_volumes_EPA_tnorm = list()
# for(i in seq_along(vol_fitted_EPA_tnorm$age)){
#  age_vol_EPA = tibble(vol_type = paste0("EPA_",vol_fitted_EPA_tnorm$age[i]),
#                       vol_L =  rnormTrunc(
#                         n = nrow(sim_conc_pathogens),
#                         mean = vol_fitted_EPA_tnorm$mean.vol[i],
#                         sd = vol_fitted_EPA_tnorm$sd.vol[i],
#                         min = vol_fitted_EPA_tnorm$lower.vol[i],
#                         max = vol_fitted_EPA_tnorm$upper.vol[i]
#                       ) / 1000,
#                       in.factor =  rnormTrunc(
#                         n = nrow(sim_conc_pathogens),
#                         mean = vol_fitted_EPA_tnorm$mean.fac[i],
#                         sd = vol_fitted_EPA_tnorm$sd.fac[i],
#                         min = vol_fitted_EPA_tnorm$lower.fac[i],
#                         max = vol_fitted_EPA_tnorm$upper.fac[i]
#                       )) %>%
#    bind_cols(sim_conc_pathogens)
#  sim_volumes_EPA_tnorm[[i]] <- age_vol_EPA
# }
# sim_volumes_EPA_tnorm <- do.call(rbind,sim_volumes_EPA_tnorm)
```

Aggregated estimation of ingestion of water, WHO (2017)

In the code below a uniform probability distribution function is assumed to sample the ingestion volume of water, following common use of uniform PDF and assumption of 1-2L ppd range (WHO 2017). Included the same `IntakeFactor` data as in the above estimations.

```
# sim_volumes_WHO = tibble(vol_type = "WHO_all", vol_L =
#                            (runif(
#                              nrow(sim_conc_pathogens), min = 1, max = 2)),
#                          in.factor =  rnormTrunc(
#                              n = nrow(sim_conc_pathogens),
#                              mean = vol_fitted_EPA_tnorm$mean.fac[i],
#                              sd = vol_fitted_EPA_tnorm$sd.fac[i],
#                              min = vol_fitted_EPA_tnorm$lower.fac[i],
#                              max = vol_fitted_EPA_tnorm$upper.fac[i]
#                            ))  %>%
#     bind_cols(sim_conc_pathogens)
```

In the code below a gamma PDF is used to estimate volumes (ml) of ingested water in Cajibio. The data to estimate this PDF comes from the self-reported data of water ingestion per day shared by community leaders in Cajibio. The Gamma distribution was estimated

at the beginning of this code. Included the same `IntakeFactor` data as in the above estimations.

```
# #volume consumed (Cajibio)
# sim_volumes_pVolCaj = tibble(
#   vol_type = "pVolCaj",
#   vol_L = (rgamma(
#     nrow(sim_conc_pathogens),
#     shape = fit.gamma$estimate[1],
#     rate = fit.gamma$estimate[2]
#   ) / 1000),
#   in.factor =  rnormTrunc(
#   n = nrow(sim_conc_pathogens),
#   mean = vol_fitted_EPA_tnorm$mean.fac[i],
#   sd = vol_fitted_EPA_tnorm$sd.fac[i],
#   min = vol_fitted_EPA_tnorm$lower.fac[i],
#   max = vol_fitted_EPA_tnorm$upper.fac[i]
# ))%>%
#     bind_cols(sim_conc_pathogens)
```

Joining the volume of ingested water simulations into one dataframe

```
# df_simulation_1 = rbind(sim_volumes_EPA_tnorm,sim_volumes_WHO,sim_volumes_pVolCaj)
#
#
# df_simulation_1 %>%
#   summarise(count = n(),
#             min.vol= min(vol_L),
#             mean.vol = mean(vol_L),
#             median.vol = mean(vol_L),
#             p95.vol = quantile(vol_L,0.95),
#             p5.vol = quantile(vol_L,0.05),
#             max.vol= max(vol_L),
#             .by =vol_type)
```

Age segregated estimations of ingestion of water

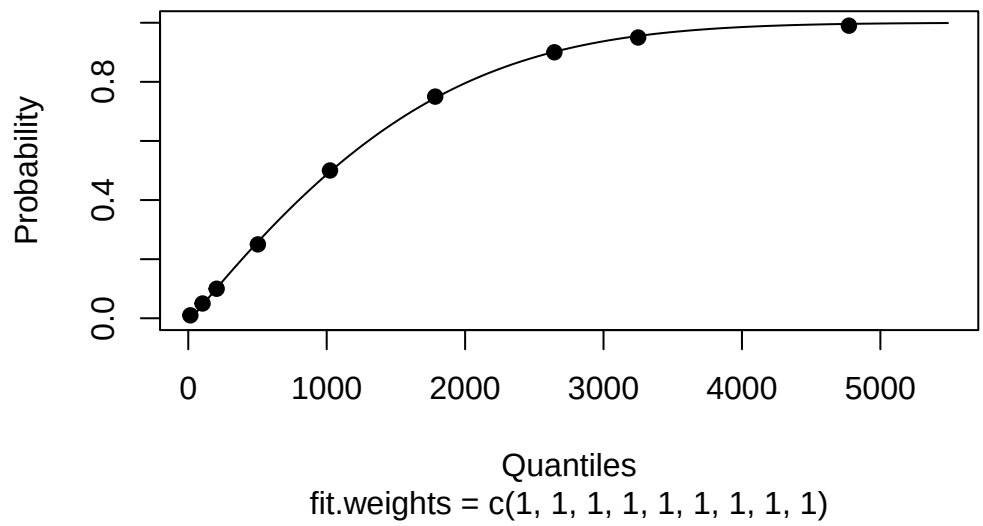
The code below is implemented to fit log-normal distributions to water consumption data (from EPA exposure Handbook 2011). The reported volumes of ingested water are organised per percentiles in individual vectors to then obtain means and standard deviations, per age group, from the fitted distribution

```
#creating a vector with percentiles
p=c(0.01,0.05,0.10,0.25,0.5,0.75,0.90,0.95,0.99)
```

```r
# #truncated normal pdf to estimate ingestion of water age 16 to 70 years
vol_fitted_tnorm_70=get.tnorm.par(p = p,
                   q = c(15,    103,    205,    503,    1024,   1784,   2645,   3250,   47
```

```
Warning: The fitting procedure 'L-BFGS-B' has failed (convergence error
occurred or specified tolerance not achieved)!

The fitting procedure 'Nelder-Mead' was successful!
(Used this fallback optimization method because 'L-BFGS-B' has failed...)

$par
[1] -601.56319 1802.49549   24.13796 9192.98249

$value
[1] 3.855001e-06

$counts
function gradient
     477       NA

$convergence
[1] 0

$message
NULL
```

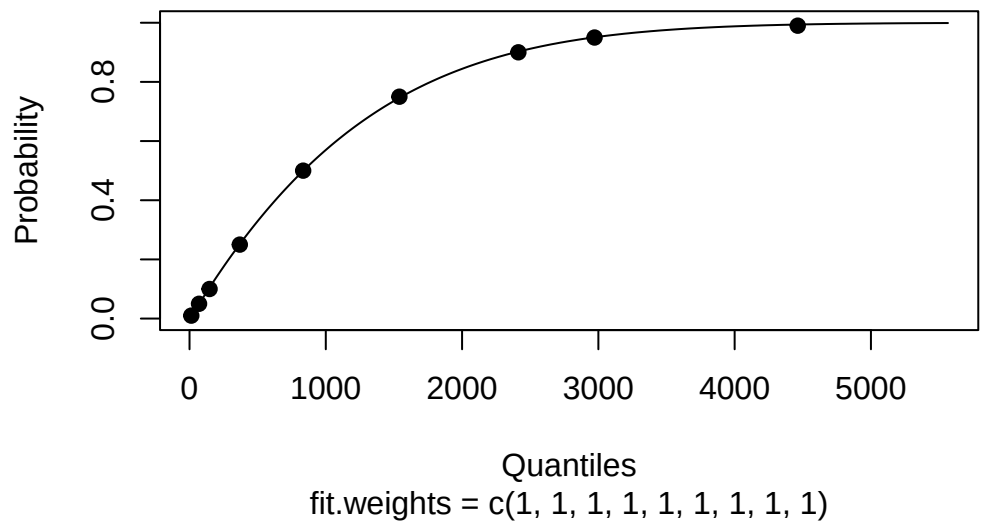
normal (mean = −601.56, sd = 1802.5, lower = 24.14, upper



```r
#truncated normal pdf to estimate ingestion of water all ages
vol_fitted_tnorm_all=get.tnorm.par(p = p,
                  q = c(13, 70, 147,    369,    834,    1540,   2413,   2972,   4463))
```

```
Warning: The fitting procedure 'L-BFGS-B' has failed (convergence error
occurred or specified tolerance not achieved)!

The fitting procedure 'Nelder-Mead' was successful!
(Used this fallback optimization method because 'L-BFGS-B' has failed...)

$par
[1] -2882.262020  2274.835068     4.797037 11460.142813

$value
[1] 9.694903e-07

$counts
function gradient
     221       NA

$convergence
[1] 0

$message
NULL
```

ıormal (mean = −2882.26, sd = 2274.84, lower = 4.8, upper :



```
# vol_fitted_lnorm_all = get.lnorm.par(p = c(.1,.25,.5,.75,.9,.95,.99),
#                           q = c(296,585,1047,1642,2345,2923,4808))
#
# vol_fitted_lnorm_21= get.lnorm.par(p = c(.1,.25,.5,.75,.9,.95,.99),
#                           q = c(506,829,1282,1877,2559,3195,5175))
#
# vol_fitted_lnorm_65= get.lnorm.par(p = c(.1,.25,.5,.75,.9,.95,.99),
#                           q = c(651,939,1345,1833,2324,2708,3750))
```

A dataframe is created to store fitted means and standard deviations

```
vol_fitted_EPA_tnorm <- tibble(age = c(
                                    "16_70",
                                    "All"),
                          bind_rows(
                            vol_fitted_tnorm_70,
                            vol_fitted_tnorm_all)
                          )


# vol_fitted_EPA <- tibble(
#   age = c(
#           "16_70",
#           "all"),
#   bind_rows(
#             vol_fitted_lnorm_21,
#             vol_fitted_lnorm_65,
#             vol_fitted_lnorm_all)
#   )
```

Values of volume of water consumed are sampled from distributions defined by the parameters in `vol_fitted_EPA` by age group

```
sim_volumes_EPA_tnorm = list()
for(i in seq_along(vol_fitted_EPA_tnorm$age)){
 age_vol_EPA = tibble(vol_type = paste0("EPA_",vol_fitted_EPA_tnorm$age[i]),
                      vol_L =  rnormTrunc(
                        n = nrow(sim_conc_pathogens),
                        mean = vol_fitted_EPA_tnorm$mean[i],
                        sd = vol_fitted_EPA_tnorm$sd[i],
                        min = vol_fitted_EPA_tnorm$lower[i],
                        max = vol_fitted_EPA_tnorm$upper[i]
                      ) / 1000
                      ) %>%
```

```
   bind_cols(sim_conc_pathogens)
 sim_volumes_EPA_tnorm[[i]] <- age_vol_EPA
}
sim_volumes_EPA_tnorm <- do.call(rbind,sim_volumes_EPA_tnorm)

# #Volume of consumed water in litres
# sim_volumes_EPA = list()
# for(i in seq_along(vol_fitted_EPA$age)){
#  age_vol_EPA = tibble(vol_type = paste0("EPA_",vol_fitted_EPA$age[i]),
#                       vol_L = rlnorm(nrow(sim_conc_pathogens),
#                       vol_fitted_EPA$meanlog[i],
#                       vol_fitted_EPA$sdlog[i])/1000) %>% bind_cols(sim_conc_pathogens)
#  sim_volumes_EPA[[i]] <- age_vol_EPA
# }
# sim_volumes_EPA <- do.call(rbind,sim_volumes_EPA)
```

Composite estimation of water ingestion

In the code below a gamma PDF is used to estimate volumes (ml) of ingested water. The data to estimate this PDF comes from the self-reported data of water ingestion per day shared by community leaders in Cajibio. The Gamma distribution was estimated at the beginning of this code.

```
#volume consumed (Cajibio)
sim_volumes_pVolCaj = tibble(
  vol_type ="pVolCaj",
  vol_L=rgamma(nrow(sim_conc_pathogens),
      shape = fit.gamma$estimate[1],rate = fit.gamma$estimate[2])/1000) %>%
  bind_cols(sim_conc_pathogens)
```

For comparison, a uniform probability distribution function is added, assuming an ingestion volume of 1-2 lppd of water (WHO 2017)

```
#volume consumed (WHO litres of water per day)
sim_volumes_WHO = tibble(vol_type = "WHO_all",
#volume of consumed water for adults, range given by WHO(2017)
                         vol_L = runif(nrow(sim_conc_pathogens), min = 1, max = 2)) %>% bi
```

Joining the volume of ingested water simulations into one dataframe

```
df_simulation_1 = rbind(sim_volumes_EPA_tnorm,sim_volumes_pVolCaj,sim_volumes_WHO)


df_simulation_1 %>%
```

```
  summarise(count = n(),
            min.vol= min(vol_L),
            mean.vol = mean(vol_L),
            median.vol = mean(vol_L),
            p95.vol = quantile(vol_L,0.95),
            p5.vol = quantile(vol_L,0.05),
            max.vol= max(vol_L),
            .by =vol_type)
```

```
# A tibble: 4 x 8
  vol_type  count min.vol mean.vol median.vol p95.vol p5.vol max.vol
  <chr>     <int>   <dbl>    <dbl>      <dbl>   <dbl>  <dbl>   <dbl>
1 EPA_16_70 60000 0.0242      1.26       1.26    3.16 0.111     7.45
2 EPA_All   60000 0.00480     1.09       1.09    2.95 0.0705    9.97
3 pVolCaj   60000 0.137       1.89       1.89    3.37 0.791     7.03
4 WHO_all   60000 1.00        1.50       1.50    1.95 1.05      2.00
```

Dose calculations

Volume ingested water and pathogen concentrations are used to calculate the dose ingested

```
# use this one when changing to in.factor mode mutate(dose = conc_p*(vol_L*in.factor))

df_dose <- df_simulation_1 %>%
  mutate(dose = conc_p*vol_L)

df_dose %>%
  summarise(count = n(),
            min.dose= min(dose),
            mean.dose = mean(dose),
            median.dose = mean(dose),
            p95.dose = quantile(dose,0.95),
            p5.dose = quantile(dose,0.05),
            max.dose= max(dose),
            .by =pathogen)
```

```
# A tibble: 3 x 8
  pathogen count min.dose   mean.dose median.dose p95.dose     p5.dose max.dose
  <chr>    <int>    <dbl>       <dbl>       <dbl>    <dbl>       <dbl>    <dbl>
1 ETEC     80000 2.77e-13 1451880045. 1451880045.  158115. 0.000000105  2.28e13
2 campy    80000 6.44e-13 5226908451. 5226908451.  901690. 0.000000546  1.34e14
3 rota     80000 1.40e-12 9376054719. 9376054719. 1034004. 0.000000705  1.47e14
```

Dose-response assessment

The probability of infection (daily) given a dose of pathogen (ETEC, Campy, Rota), is estimated using a Beta-Poisson model

$$risk = 1 - \left[1 + dose\frac{2^{1/a} - 1}{N50}\right]^{-a}$$

In the code below the point/range/PDF estimates for a and $N50$ are defined for the pathogens of study (ETEC, Campy and rota)

```
df_dose_resp <- df_dose %>%
  mutate(N50 = case_when(
             pathogen == "ETEC" ~ 1.7e06,#from Moncada-Barragan (2021)
             #pathogen == "campy" ~ rlnorm(nrow(df_dose), 1.69e03, 2.78e03),#from Bivins et
             pathogen == "campy" ~ 6.68e4,#from QMRA wiki
             pathogen == "rota" ~ rlnorm(nrow(df_dose), 8.16, 6.65)),#from Bivins et al (20
               #pathogen == "rota" ~ 6.17),#from QMRA wiki
         a = case_when(
             pathogen == "ETEC" ~ 0.0754,#from Moncada-Barragan (2021)
             #pathogen == "campy" ~ rlnorm(nrow(df_dose), 1.51e-01, 5.90e-02),#from Bivins
             pathogen == "campy" ~ 3.19e-01, #from QMRA wiki
             pathogen == "rota" ~ rlnorm(nrow(df_dose), 2.48e-01, 1.46e-1))) #from Bivins e
             #pathogen == "rota" ~2.5e-02)) #from QMRA wiki
```

Risk characterisation

Daily infection risk

In the code below, I use the Beta-Poisson equation to estimate probability of infection given an ingested dose. All results are stored in an appended dataframe, df_risk.

```
df_risk <- df_dose_resp %>%
  mutate( risk = 1-(1+(dose/N50)*((2^(1/a)-1)))^-a)

df_risk %>%
  summarise(count = n(),
            min.risk= min(risk),
            mean.risk = mean(risk),
            median.risk = mean(risk),
            p95.risk = quantile(risk,0.95),
            p5.risk = quantile(risk,0.05),
            max.risk= max(risk),
            .by =c(vol_type, pathogen, w_source))
```

```
# A tibble: 24 x 10
   vol_type  pathogen w_source count min.risk mean.risk median.risk  p95.risk
   <chr>     <chr>    <chr>    <int>    <dbl>     <dbl>       <dbl>     <dbl>
 1 EPA_16_70 ETEC     tap      10000 2.22e-16  0.000419    0.000419 0.000114
 2 EPA_16_70 campy    tap      10000 0         0.000434    0.000434 0.0000601
 3 EPA_16_70 rota     tap      10000 0         0.0342      0.0342   0.126
 4 EPA_16_70 ETEC     stored   10000 6.55e-15  0.104       0.104    0.500
 5 EPA_16_70 campy    stored   10000 1.11e-15  0.168       0.168    0.897
 6 EPA_16_70 rota     stored   10000 0         0.344       0.344    1.00
 7 EPA_All   ETEC     tap      10000 1.11e-16  0.000329    0.000329 0.0000942
 8 EPA_All   campy    tap      10000 1.11e-16  0.000456    0.000456 0.0000496
 9 EPA_All   rota     tap      10000 0         0.0296      0.0296   0.0799
10 EPA_All   ETEC     stored   10000 2.11e-14  0.0995      0.0995   0.495
# i 14 more rows
# i 2 more variables: p5.risk <dbl>, max.risk <dbl>
```

Yearly infection risk

In the code below, I calculate yearly infection risk using the the previously defined daily infection risk. All results are stored in an appended dataframe.

```r
df_yearly_risk <- df_risk %>%
  mutate(yearly_risk = 1-(1-risk)^365)

df_yearly_risk %>%
  summarise(count = n(),
            min.riskyear= min(yearly_risk),
            mean.riskyear = mean(yearly_risk),
            median.riskyear = mean(yearly_risk),
            p95.riskyear = quantile(yearly_risk,0.95),
            p5.riskyear = quantile(yearly_risk,0.05),
            max.riskyear= max(yearly_risk),
            .by =pathogen)
```

```
# A tibble: 3 x 8
  pathogen count min.riskyear mean.riskyear median.riskyear p95.riskyear
  <chr>    <int>        <dbl>         <dbl>           <dbl>        <dbl>
1 ETEC     80000     4.05e-14         0.290           0.290            1
2 campy    80000     0                0.268           0.268            1
3 rota     80000     0                0.367           0.367            1
# i 2 more variables: p5.riskyear <dbl>, max.riskyear <dbl>
```

Daily illness risk

Finally, we calculate illness risk multiplying daily infection risk by the Colombia's ADD (acute diarrheal disease) morbidity rate (reported by the INS in 2023).

```
df_illness_risk <- df_risk %>%
  mutate(ill_risk = risk * morbidity)

df_illness_risk %>%
  summarise(count = n(),
            min.riskIll= min(ill_risk),
            mean.riskIll = mean(ill_risk),
            median.riskIll = mean(ill_risk),
            p95.riskIll = quantile(ill_risk,0.95),
            p5.riskIll = quantile(ill_risk,0.05),
            max.riskIll= max(ill_risk),
            .by =pathogen)
```

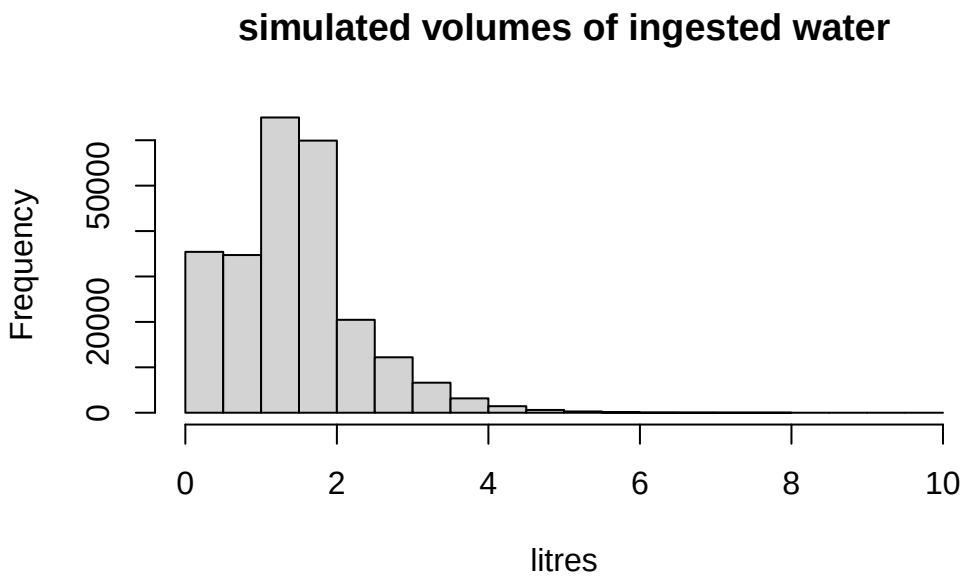
```
# A tibble: 3 x 8
  pathogen count min.riskIll mean.riskIll median.riskIll p95.riskIll p5.riskIll
  <chr>    <int>       <dbl>        <dbl>          <dbl>       <dbl>      <dbl>
1 ETEC     80000    2.22e-17       0.0108         0.0108      0.0804   9.12e-12
2 campy    80000    0              0.0176         0.0176      0.155    4.06e-12
3 rota     80000    0              0.0384         0.0384      0.200    3.82e-13
# i 1 more variable: max.riskIll <dbl>
```

Visualisation of results

Volume of ingested water

Below, an histogram of the overall simulated volumes of ingested water

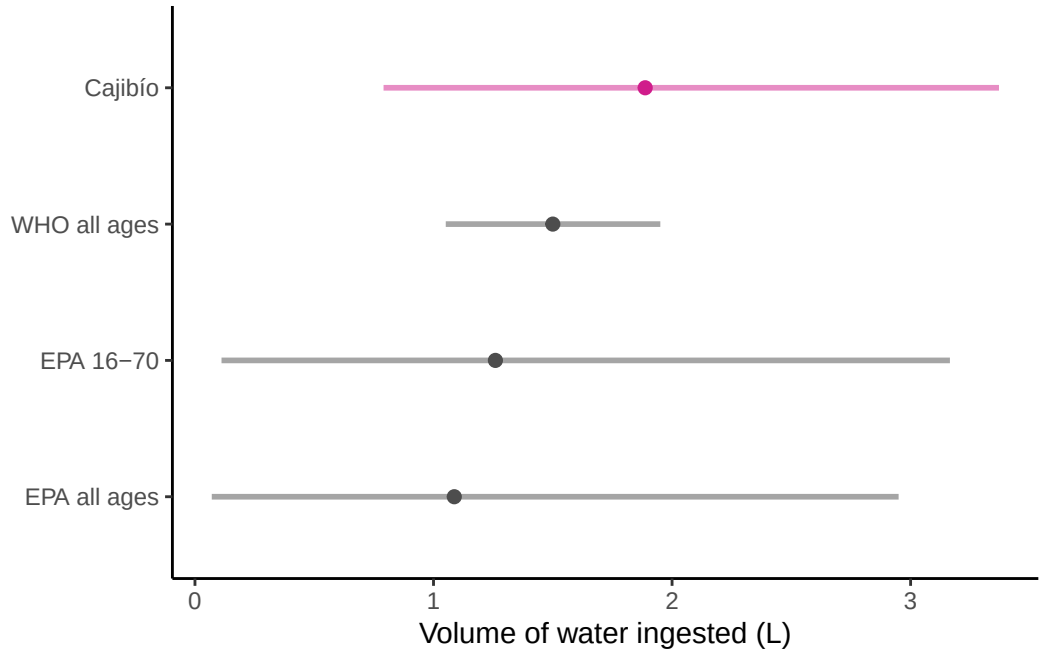
```
hist(df_risk$vol_L, main="simulated volumes of ingested water", xlab="litres")
```



Through the code below we visualize the simulation of volumes of ingested water, while using the WHO (2017) values, sampled using uniform distribution; and the EPA (2019) values, sampled using log normal distributions.

```
vol_ing <- df_risk %>%
  summarise(count = n(),
            mean.vol_L = mean(vol_L),
            median.vol_L = mean(vol_L),
            p95.vol_L = quantile(vol_L,0.95),
            p5.vol_L = quantile(vol_L,0.05),
            .by = vol_type) %>%
  mutate(vol_type=case_when(
    vol_type=="EPA_All"~"EPA all ages",
    vol_type=="EPA_16_70"~"EPA 16-70",
    vol_type=="pVolCaj"~"Cajibío",
    vol_type=="WHO_all"~"WHO all ages"
    )) %>%
  ggplot(aes(x=fct_reorder(vol_type, median.vol_L, mean)))+
  geom_linerange(aes(ymin = p5.vol_L,ymax = p95.vol_L, col=vol_type),
                 lwd = 1,alpha = 0.5)+
  geom_point(aes(y=mean.vol_L, col=vol_type), size=2)+
  coord_flip()+
  scale_color_manual(values=c("#d01c8b","gray30", "gray30","gray30"))+
  theme_classic()+
  labs(x = NULL ,
       y = "Volume of water ingested (L)")+
```

```
  theme(legend.position = "none")


print(vol_ing)
```



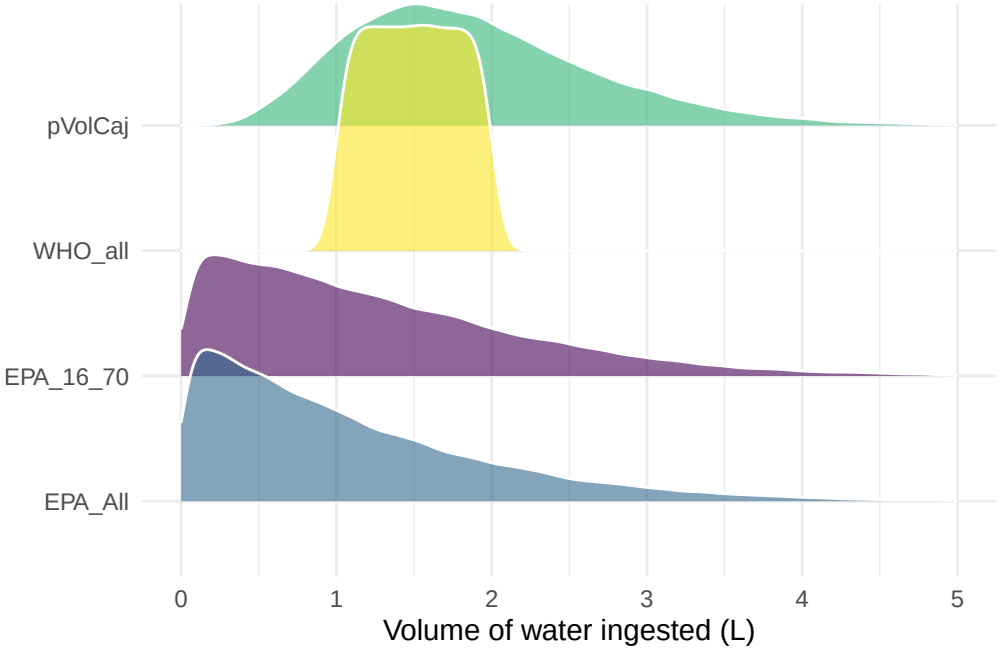

```
ggsave("vol_ing_pQMRA.png", plot = vol_ing, dpi = 300, units = "cm", width = 15, height =
```

The code below shows the same results as above, the only difference is the display of the distributions.

```
df_risk %>%
  ggplot(aes(y=fct_reorder(vol_type,vol_L,mean), x=vol_L,fill=vol_type))+
  geom_density_ridges(alpha = 0.6,col = "white")+
  scale_x_continuous(limits = c(0,5))+
  scale_fill_viridis_d()+
  theme_minimal()+
  labs(y="",x="Volume of water ingested (L)")+
  theme(legend.position = "none")
```

```
Picking joint bandwidth of 0.0724
```

```
Warning: Removed 450 rows containing non-finite outside the scale range
(`stat_density_ridges()`).
```

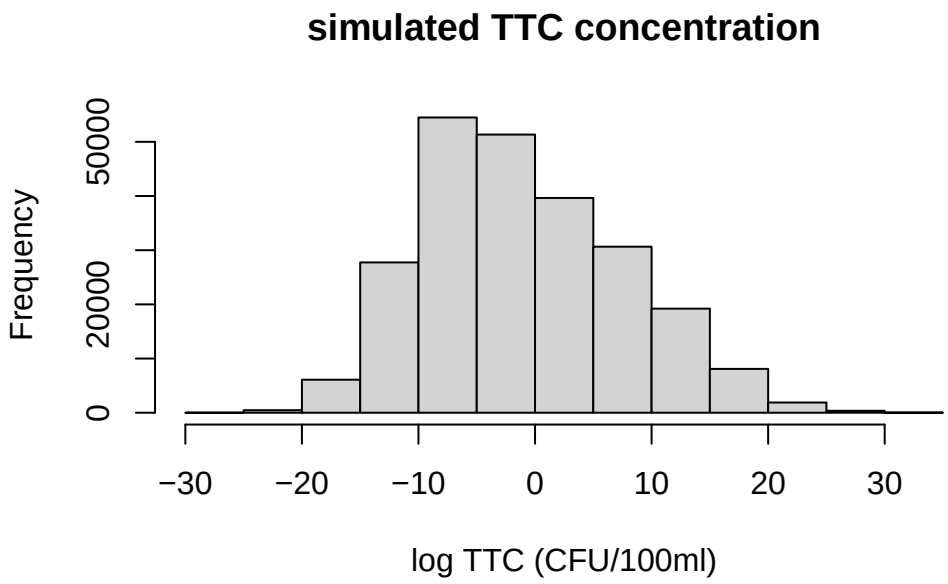


Concentration of pathogens

Thermotolerant coliforms

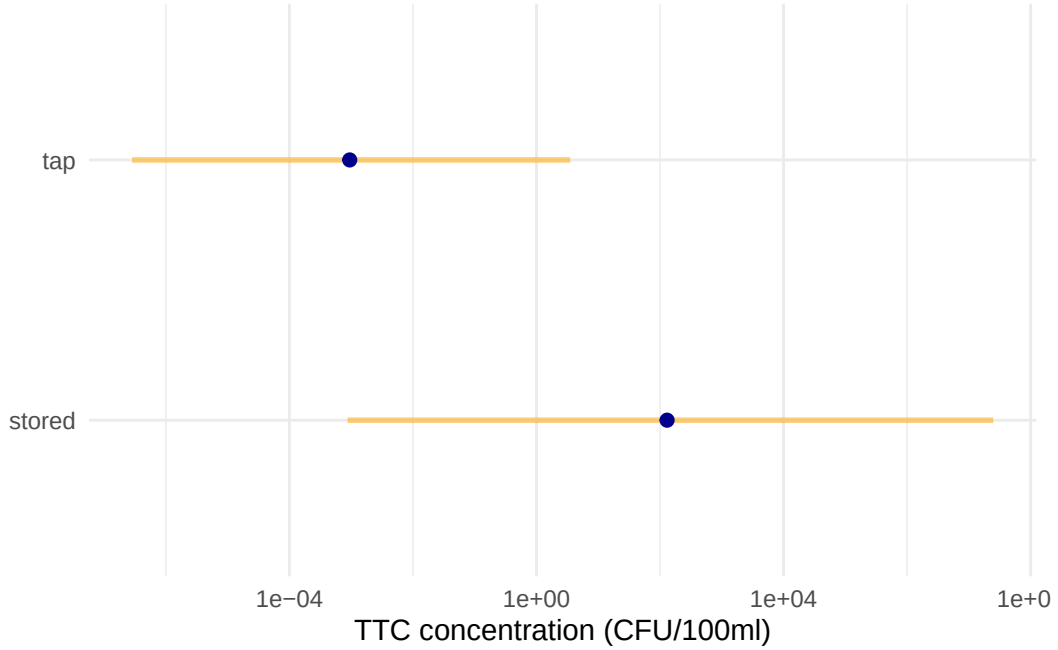
Below, an histogram of the overall simulated TTC concentrations. The values were sampled using the MLE estimates.

```
hist(log(df_risk$conc_0), main = "simulated TTC concentration", xlab = "log TTC (CFU/100ml
```



Through the code below we visualize the simulation of concentration of TTCs in stored and tap water.

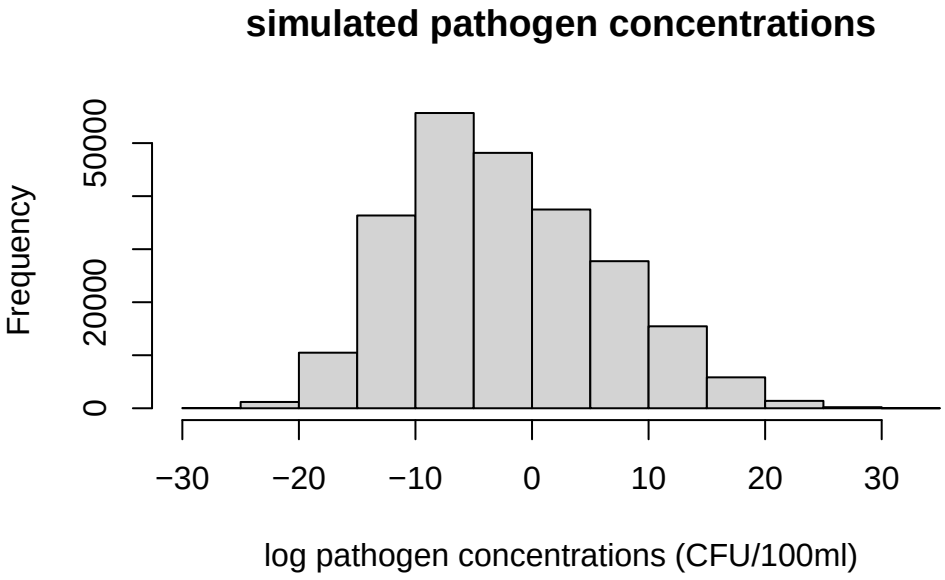
```
  df_risk %>%  summarise(across(conc_0,
                                list(mean = mean,
                                     median = median,
                                     p5 = \(x) quantile(x,0.05),
                                     p95 = \(x) quantile(x,0.95)
                                     )
                                ),
                         .by = w_source
                         ) |>
  ggplot(aes(x = w_source))+
  geom_linerange(aes(ymin = conc_0_p5,ymax = conc_0_p95),
                 col="orange", lwd = 1,alpha = 0.5)+
  geom_point(aes(y=conc_0_median), col="darkblue", size=2)+
  coord_flip()+
  scale_y_log10()+
  theme_minimal()+
labs(x = NULL , y = "TTC concentration (CFU/100ml)")
```



Enteropathogens

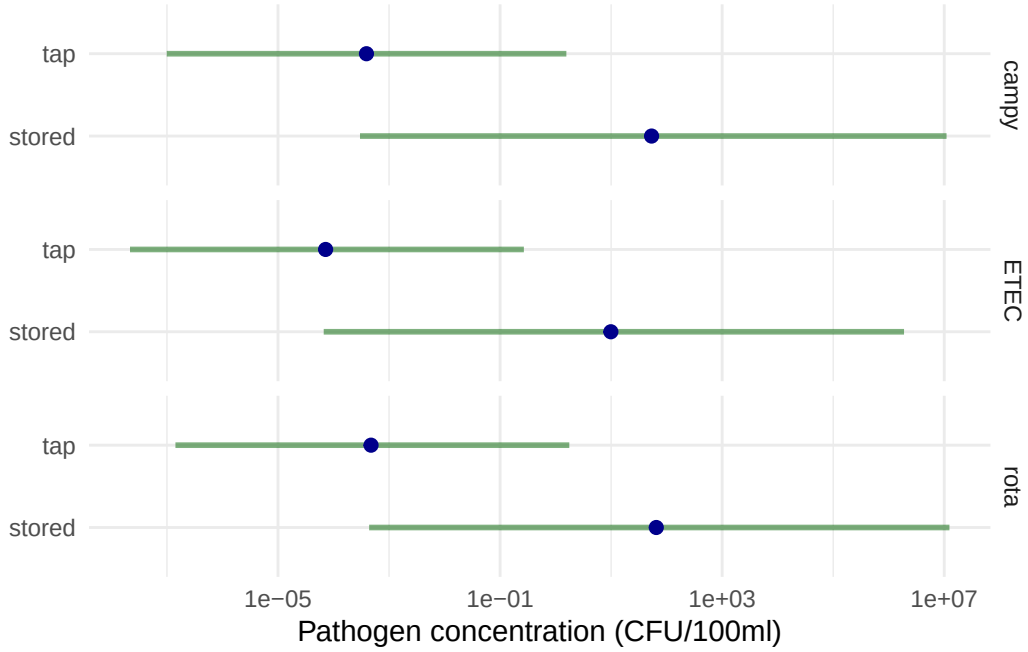
Below, an histogram of the overall simulated pathogen concentrations. The values were sampled using the MLE estimates obtained previously.

```
hist(log(df_risk$conc_p), main="simulated pathogen concentrations", xlab = "log pathogen c
```



Through the code below we visualise the simulation of concentration of three pathogens (ETEC, Campylobacter jejuni, and rotavirus), in sampled tap and stored water.

```
path_conc <-   df_risk |> summarise(across(conc_p,
                                list(mean = mean,
                                     median = median,
                                     p5 = \(x) quantile(x,0.05),
                                     p95 = \(x) quantile(x,0.95)
                                     )
                                ),
                        .by = c(w_source, pathogen)
                        ) |>
  ggplot(aes(x = w_source))+
  geom_linerange(aes(ymin = conc_p_p5,ymax = conc_p_p95),
                 col="darkgreen", lwd = 1,alpha = 0.5)+
  geom_point(aes(y=conc_p_median), col="darkblue", size=2)+
  coord_flip()+
  scale_y_log10()+
  theme_minimal()+
  facet_grid(pathogen~.)+
  labs(x = NULL , y = "Pathogen concentration (CFU/100ml)")

print(path_conc)
```



```
 df_risk |> summarise(across(conc_p,
                              list(mean = mean,
                                   median = median,
                                   p5 = \(x) quantile(x,0.05),
                                   p95 = \(x) quantile(x,0.95)
                                   )
                              ),
                          .by = c(w_source, pathogen))
```

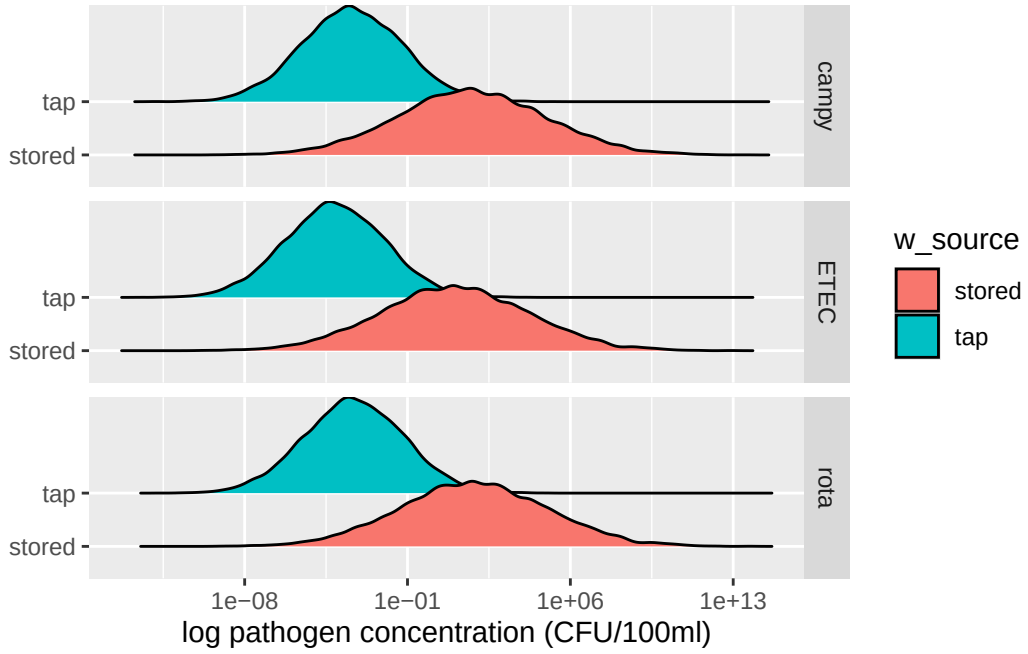
```
# A tibble: 6 x 6
  w_source pathogen conc_p_mean conc_p_median    conc_p_p5   conc_p_p95
  <chr>    <chr>          <dbl>         <dbl>        <dbl>        <dbl>
1 tap      ETEC         4.13e 0     0.0000720 0.0000000216        0.268
2 tap      campy        2.95e 1     0.000392  0.0000000991        1.56
3 tap      rota         2.71e 1     0.000474  0.000000142         1.76
4 stored   ETEC         1.79e 9     9.89      0.0000666     1886444.
5 stored   campy        7.23e 9    53.5       0.000300     11012491.
6 stored   rota         1.18e10    65.0       0.000438     12410827.
```

The code below shows the same results as above, the only difference is the display of the distributions.

```
df_risk |>
  ggplot(aes(x = conc_p,
             y = w_source,
             fill = w_source
             ))+
  geom_density_ridges()+
  scale_x_log10()+
  facet_grid(pathogen~.)+
  labs(y = NULL , x = "log pathogen concentration (CFU/100ml)")
```

```
Picking joint bandwidth of 0.29
```

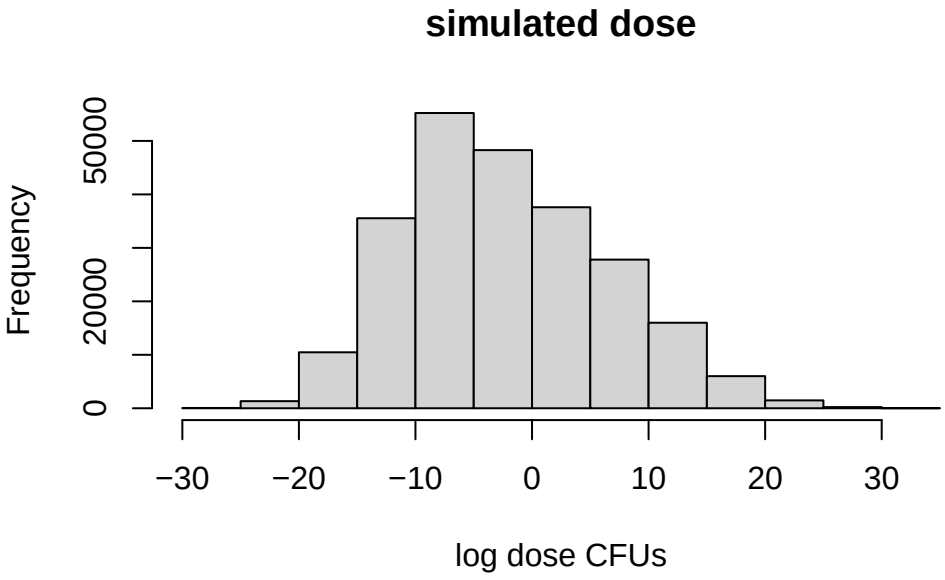
```
Picking joint bandwidth of 0.288
Picking joint bandwidth of 0.288
```



Dose of pathogens

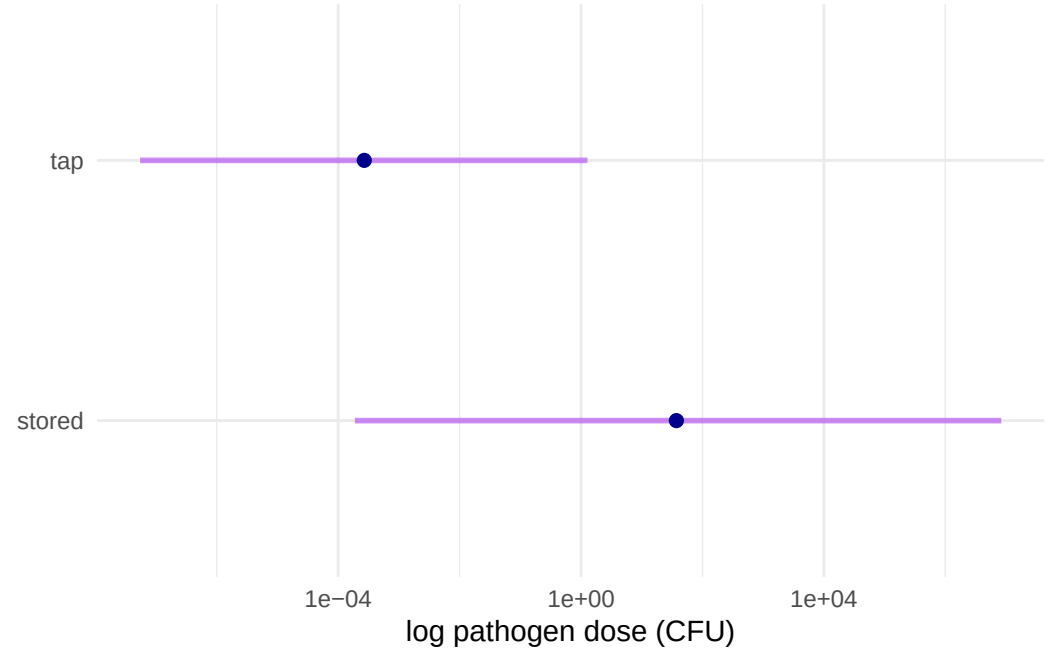
Below, an histogram of the overall simulated dose. The values were obtained from multiplying the simulations of concentration of pathogens and volume of water consumed.

```
hist(log(df_risk$dose), main = "simulated dose", xlab = "log dose CFUs")
```



Through the code below we visualize the dose simulation in stored and tap water.

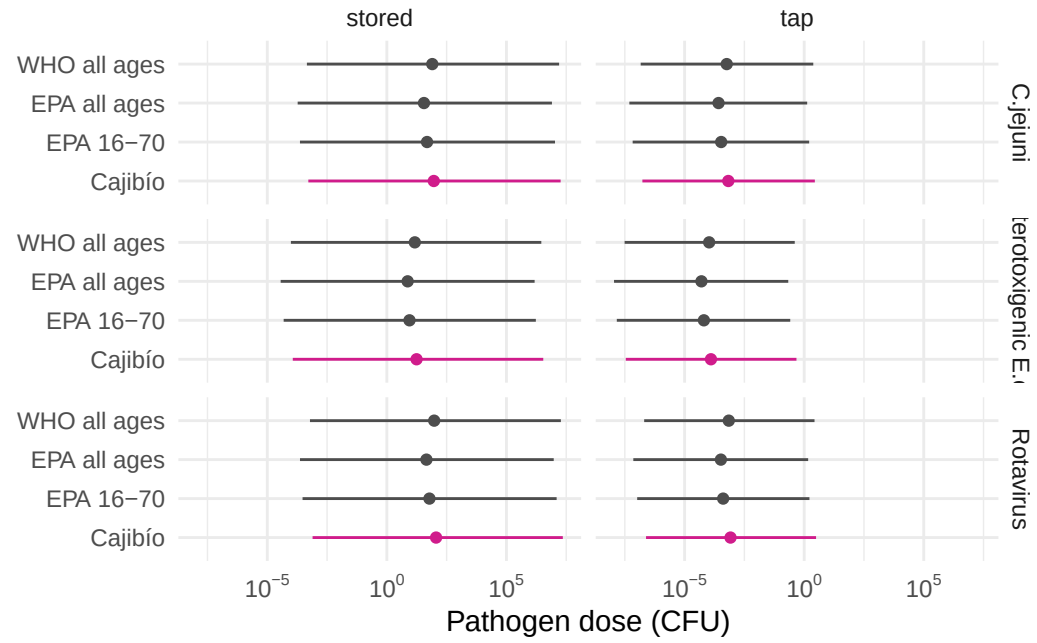
```
  df_risk %>%  summarise(across(dose,
                               list(mean = mean,
                                    median = median,
                                    p5 = \(x) quantile(x,0.05),
                                    p95 = \(x) quantile(x,0.95)
                                    )
                               ),
                         .by = w_source
                         ) |>
  ggplot(aes(x = w_source))+
  geom_linerange(aes(ymin = dose_p5,ymax = dose_p95),
                 col="purple", lwd = 1,alpha = 0.5)+
  geom_point(aes(y=dose_median), col="darkblue", size=2)+
  coord_flip()+
  scale_y_log10()+
  theme_minimal()+
labs(x = NULL , y = "log pathogen dose (CFU)")
```



Through the code below we visualize the results for dose calculation per type of water sample and pathogens

```
path_dose <-   df_risk |> summarise(across(dose,
                               list(mean = mean,
                                    median = median,
```

```
                                  p5 = \(x) quantile(x,0.05),
                                  p95 = \(x) quantile(x,0.95)
                                  )
                              ),
                      .by = c(vol_type, w_source, pathogen)
                      ) %>%
    mutate(vol_type=case_when(
    vol_type=="EPA_All"~"EPA all ages",
    vol_type=="EPA_16_70"~"EPA 16-70",
    vol_type=="pVolCaj"~"Cajibío",
    vol_type=="WHO_all"~"WHO all ages"
    ),
     pathogen = case_when(
      pathogen == "ETEC" ~ "Enterotoxigenic E.coli",
      pathogen == "campy" ~ "C.jejuni",
      pathogen == "rota" ~ "Rotavirus")) %>%
  ggplot(aes(x = vol_type))+
  geom_linerange(aes(ymin = dose_p5,ymax = dose_p95, col=vol_type))+
  geom_point(aes(y=dose_median, col=vol_type))+
  facet_grid(pathogen~w_source)+
  coord_flip()+
  scale_y_log10(breaks = scales::trans_breaks
                ("log10", function(x) 10^x),
                labels = scales::trans_format
                ("log10", scales::math_format(10^.x)))+
  scale_color_manual(values=c("#d01c8b","gray30", "gray30","gray30"))+
  theme_minimal()+
  labs(x = NULL , y = "Pathogen dose (CFU)")+
  theme(legend.position = "none")

print(path_dose)
```




```
ggsave("path_dose_pQMRA.png", plot = path_dose, dpi = 300, units = "cm", width = 15, heigh
```

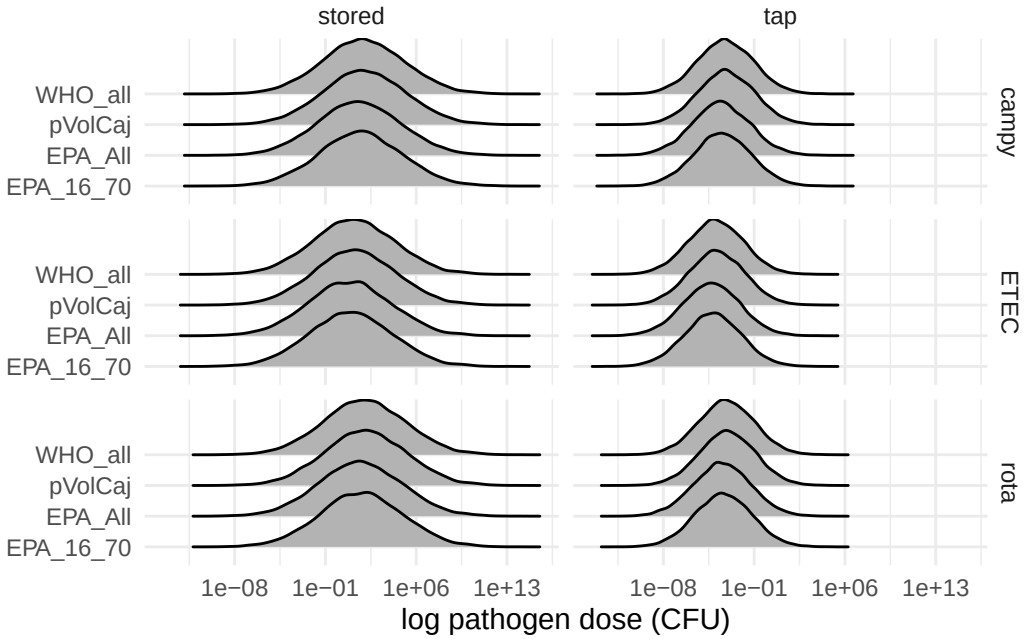
```
df_risk |> summarise(across(dose,
                            list(mean = mean,
                                 median = median,
                                 p5 = \(x) quantile(x,0.05),
                                 p95 = \(x) quantile(x,0.95)
                                 )
                            ),
                     .by = c(vol_type,w_source))
```

```
# A tibble: 8 x 6
  vol_type  w_source     dose_mean dose_median      dose_p5     dose_p95
  <chr>     <chr>            <dbl>       <dbl>        <dbl>        <dbl>
1 EPA_16_70 tap               26.3    0.000210 0.0000000408        1.05
2 EPA_16_70 stored    8079877434.     29.1      0.000136      6341156.
3 EPA_All   tap               33.2    0.000166 0.0000000301        0.856
4 EPA_All   stored    9213662360.     23.2      0.000106      5132914.
5 pVolCaj   tap               42.4    0.000421 0.0000000959        1.85
6 pVolCaj   stored   14437658797.     57.7      0.000328     12495883.
7 WHO_all   tap               29.7    0.000360 0.0000000841        1.45
8 WHO_all   stored   11081716518.     48.9      0.000258     10650352.
```

The code below shows the same results as above, the only difference is the display of the distributions.

```
df_risk |>
ggplot(aes(x = dose, y=vol_type))+
geom_density_ridges()+
facet_grid(pathogen~w_source)+
scale_x_log10()+
theme_minimal()+
labs(y = NULL , x = "log pathogen dose (CFU)")
```

```
Picking joint bandwidth of 0.457

Picking joint bandwidth of 0.316

Picking joint bandwidth of 0.456

Picking joint bandwidth of 0.312

Picking joint bandwidth of 0.456

Picking joint bandwidth of 0.312
```

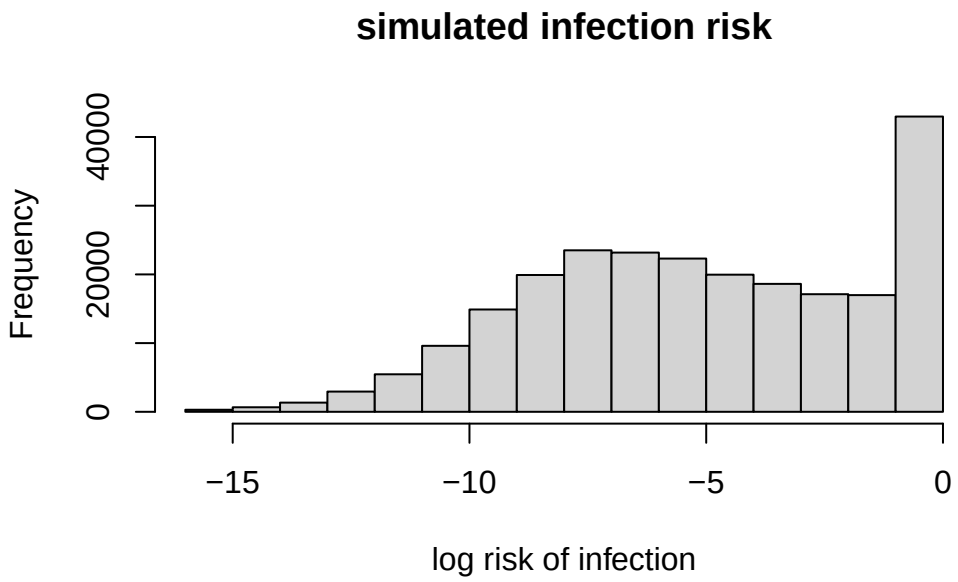


Risks

Infection

First, an overview of the overall risk simulations by plotting an histogram

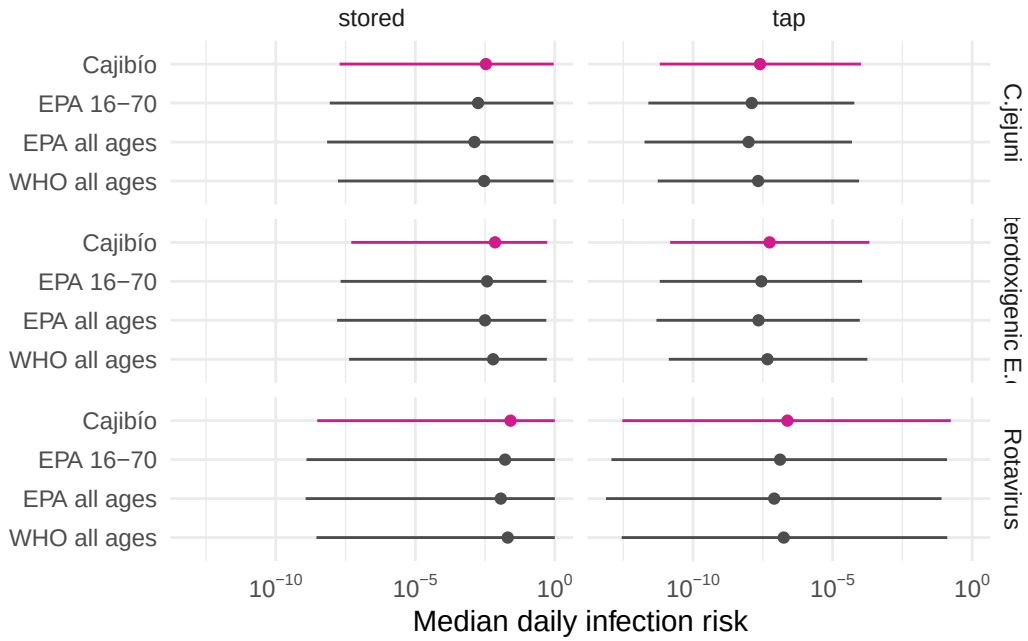
```
hist(log10(df_risk$risk), main = "simulated infection risk", xlab = "log risk of infection
```



Through the code below we visualize the results of median daily risk of infection for the 3 pathogens of study.

```
risk_daily <- df_risk |> summarise(across(
  risk,
  list(
    mean = mean,
    median = median,
    p5 = \(x) quantile(x, 0.05),
    p95 = \(x) quantile(x, 0.95)
  )
), .by = c(vol_type, w_source, pathogen)) |>
  mutate(
    vol_type = case_when(
      vol_type == "EPA_All" ~ "EPA all ages",
      vol_type == "EPA_16_70" ~ "EPA 16-70",
      vol_type == "pVolCaj" ~ "Cajibío",
      vol_type == "WHO_all" ~ "WHO all ages"
```

```
    ),
    pathogen = case_when(
      pathogen == "ETEC" ~ "Enterotoxigenic E.coli",
      pathogen == "campy" ~ "C.jejuni",
      pathogen == "rota" ~ "Rotavirus"
    )
  ) %>%

  ggplot(aes(x = fct_rev(vol_type))) +
  #geom_hline(yintercept = 1e-4,col = "#E64B35",linewidth = 1.2,alpha = 0.6,linetype = "da
  geom_linerange(aes(ymin = risk_p5, ymax = risk_p95, col = vol_type)) +
  geom_point(aes(y = risk_median, col = vol_type)) +
  facet_grid(pathogen ~ w_source) +
  coord_flip() +
  scale_y_log10(
    breaks = scales::trans_breaks("log10", function(x)
      10 ^ x),
    labels = scales::trans_format("log10", scales::math_format(10 ^
                                                                 .x))
  ) +
  scale_color_manual(values = c("#d01c8b", "gray30", "gray30", "gray30")) +
  theme_minimal() +
  labs(x = NULL, y = "Median daily infection risk") +
  theme(legend.position = "none")

print(risk_daily)
```

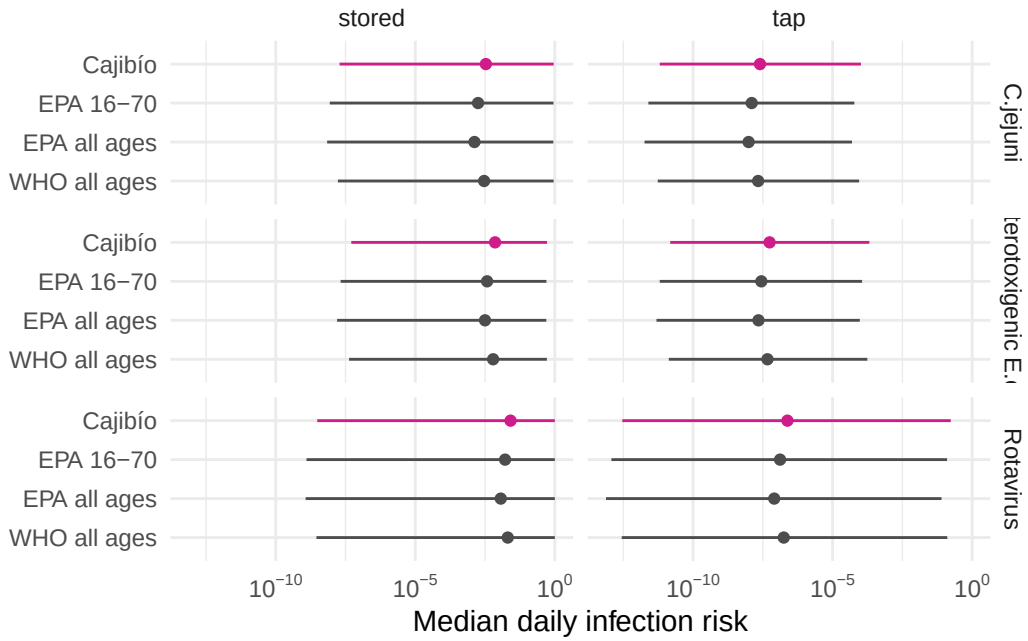



```
ggsave("risk_daily_pQMRA.png", plot = risk_daily, dpi = 330, units = "cm", width = 15, hei
```

The code below generates a visualization of the infection risk assessment, but this time the labels are in spanish and the scales reflect the probabilities in a way that was easier to use for the dissemination of results with the Cajibío community. The dashed line indicates the 10-4 benchmark of yearly infections set by the EPA red book.

```
riesgo_diario <- df_risk |> summarise(across(risk,
                              list(mean = mean,
                                   median = median,
                                   p5 = \(x) quantile(x,0.05),
                                   p95 = \(x) quantile(x,0.95)
                                   )
                              ),
                        .by = c(vol_type, w_source, pathogen)
                        ) |>
  mutate(vol_type = case_when(
    vol_type=="EPA_16_70"~"EPA Adultos",
    vol_type=="EPA_All"~"EPA Todos",
    vol_type=="pVolCaj"~"Cajibío Adultos",
    vol_type=="WHO_all"~ "OMS Todos"
    ),
    w_source = case_when(w_source=="tap"~"Llave",
                         T~"Almacenada"),
    pathogen = case_when(pathogen=="ETEC"~"Bacteria (E.coli)",
```

```
                        pathogen=="campy"~"Bacteria (C.jejuni)",
                        pathogen=="rota"~"Virus")) |>
  mutate(vol_type = factor(vol_type,
                           levels = c("Cajibío Adultos","OMS Todos","EPA Todos","EPA Adult
                           ordered = T)) |>
  ggplot(aes(x = fct_rev(vol_type)))+
  geom_hline(yintercept = 1e-4,col = "#E64B35",linewidth = 1.2,alpha = 0.6,linetype = "das
  geom_linerange(aes(ymin = risk_p5,ymax = risk_p95, col=vol_type))+
  geom_point(aes(y=risk_median, col=vol_type))+
  facet_grid(pathogen~w_source)+
  coord_flip()+
  # scale_y_log10(breaks = scales::trans_breaks("log10", function(x) 10^x),
  #               labels = scales::trans_format("log10", scales::math_format(10^.x)))+
  scale_y_log10(breaks = c(1e-10,1e-8,1e-6,1e-4,1e-2,1e0),
                labels = c("1 en 10.000 millones",
                           "1 en 100 millones",
                           "1 en 1 millon",
                           "1 en 10 mil",
                           "1 en 100",
                           "1 en 1"))+
  scale_color_manual(values=c("#00A087","#3C5488", "#3C5488","#3C5488"))+
  theme_minimal()+
  labs(x = NULL , y ="",title = "Riesgo de infección diario (mediana)")+
  theme(legend.position = "none",
        axis.text.x = element_text(angle = -90, vjust = 0.5, hjust = 1),
        panel.grid.minor = element_blank())

print(risk_daily)
```




```
ggsave("riesgo_diario_pQMRA.png", plot = risk_daily, dpi = 330, units = "cm", width = 15,
```

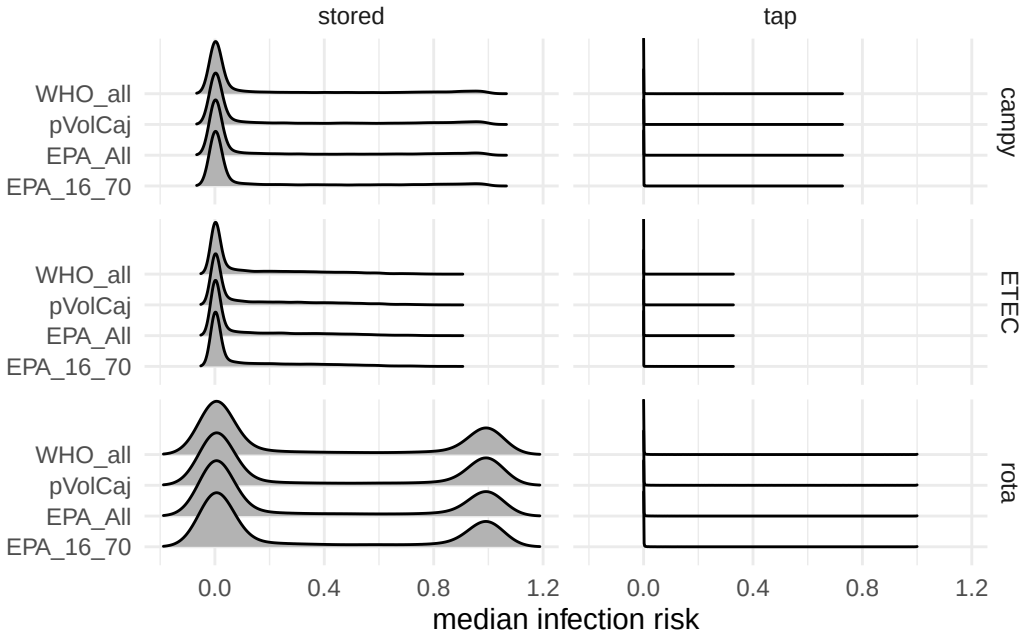
```
df_risk |> summarise(across(risk,
                            list(mean = mean,
                                 median = median,
                                 p5 = \(x) quantile(x,0.05),
                                 p95 = \(x) quantile(x,0.95)
                                 )
                            ),
                     .by = c(vol_type, w_source))
```

```
# A tibble: 8 x 6
  vol_type  w_source risk_mean  risk_median  risk_p5 risk_p95
  <chr>     <chr>        <dbl>        <dbl>    <dbl>    <dbl>
1 EPA_16_70 tap         0.0117 0.0000000299 1.20e-12  0.00188
2 EPA_16_70 stored      0.205  0.00431      6.28e- 9  0.999
3 EPA_All   tap         0.0101 0.0000000221 8.72e-13  0.00155
4 EPA_All   stored      0.198  0.00331      5.32e- 9  0.999
5 pVolCaj   tap         0.0131 0.0000000570 2.96e-12  0.00269
6 pVolCaj   stored      0.224  0.00782      1.39e- 8  1.00
7 WHO_all   tap         0.0114 0.0000000468 2.71e-12  0.00252
8 WHO_all   stored      0.218  0.00639      1.28e- 8  1.00
```

The code below shows the same results as above, the only difference is the display of the distributions.

```r
df_risk |>
ggplot(aes(x = risk, y=vol_type))+
geom_density_ridges()+
facet_grid(pathogen~w_source)+
#scale_x_log10()+
theme_minimal()+
labs(y = NULL , x = "median infection risk")
```

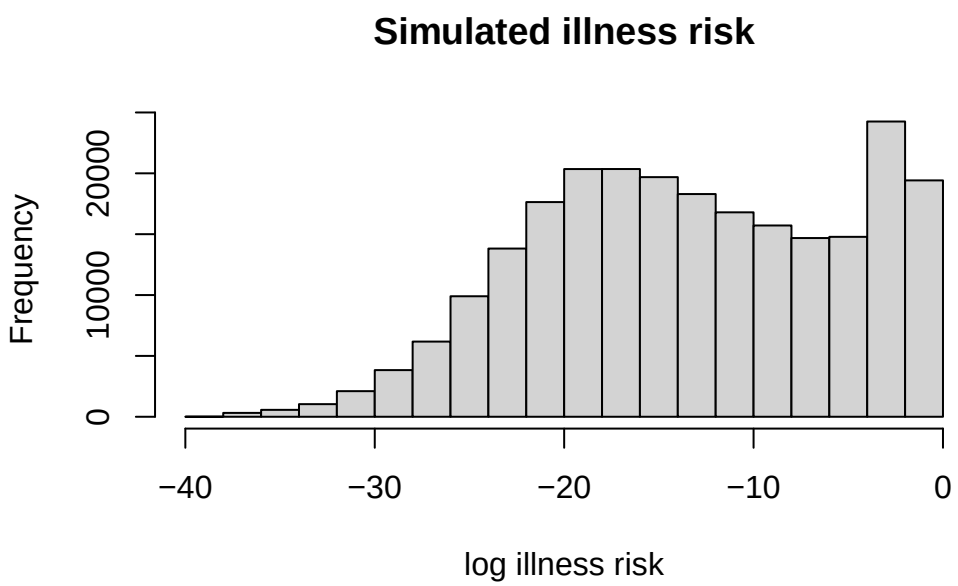
```
Picking joint bandwidth of 0.0223

Picking joint bandwidth of 6.03e-08

Picking joint bandwidth of 0.0172

Picking joint bandwidth of 1.23e-07

Picking joint bandwidth of 0.0627

Picking joint bandwidth of 4.64e-06
```



Illness

First, an overview of the overall illness risk simulations by plotting an histogram

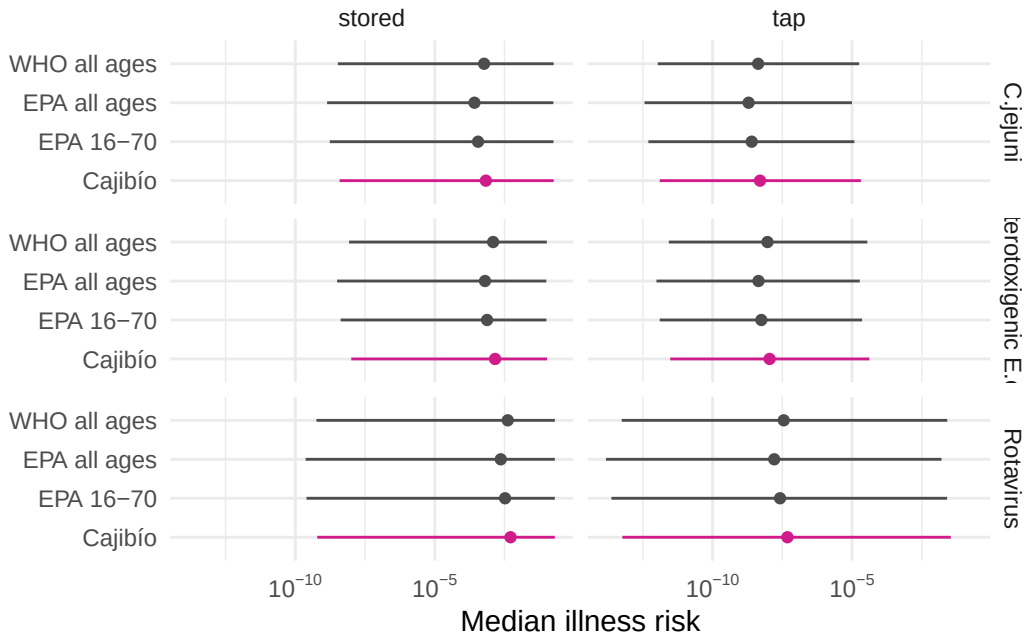
```
hist(log(df_illness_risk$ill_risk), main = "Simulated illness risk", xlab = "log illness r
```



Through the code below we visualize the results of estimating the daily risk of illness using the Colombian morbidity ratio and the beta-poisson model of the 3 pathogens of study.

```
risk_illn <- df_illness_risk |> summarise(across(
  ill_risk,
  list(
    mean = mean,
    median = median,
    p5 = \(x) quantile(x, 0.05),
    p95 = \(x) quantile(x, 0.95)
  )
), .by = c(vol_type, w_source, pathogen)) |>
  mutate(
    vol_type = case_when(
      vol_type == "EPA_All" ~ "EPA all ages",
      vol_type == "EPA_16_70" ~ "EPA 16-70",
      vol_type == "pVolCaj" ~ "Cajibío",
      vol_type == "WHO_all" ~ "WHO all ages"
    ),
    pathogen = case_when(
```

```
      pathogen == "ETEC" ~ "Enterotoxigenic E.coli",
      pathogen == "campy" ~ "C.jejuni",
      pathogen == "rota" ~ "Rotavirus"
    )
  ) %>%
  ggplot(aes(x = vol_type)) +
  geom_linerange(aes(ymin = ill_risk_p5, ymax = ill_risk_p95, col = vol_type)) +
  geom_point(aes(y = ill_risk_median, col = vol_type)) +
  facet_grid(pathogen ~ w_source) +
  coord_flip() +
  scale_y_log10(
    breaks = scales::trans_breaks
    ("log10", function(x)
      10 ^ x),
    labels = scales::trans_format
    ("log10", scales::math_format(10 ^ .x))
  ) +
  scale_color_manual(values = c("#d01c8b", "gray30", "gray30", "gray30")) +
  theme_minimal() +
  labs(x = NULL , y = "Median illness risk") +
  theme(legend.position = "none")

print(risk_illn)
```



```
ggsave("risk_illn_pQMRA.png", plot = risk_illn, dpi = 300, units = "cm", width = 15, heigh
```

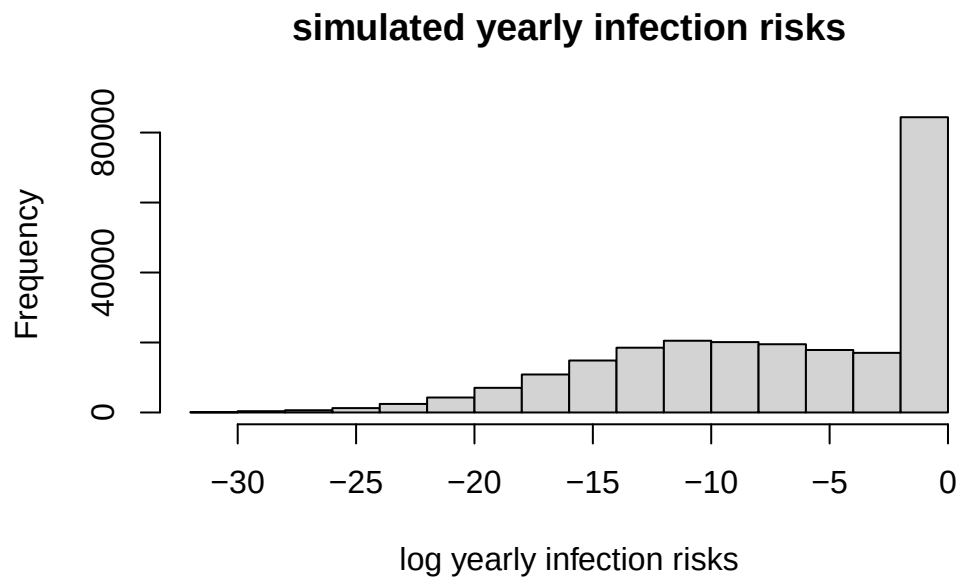
```
df_illness_risk %>%  summarise(across(ill_risk,
                               list(mean = mean,
                                    median = median,
                                    p5 = \(x) quantile(x,0.05),
                                    p95 = \(x) quantile(x,0.95)
                                    )
                               ),
                       .by = c(vol_type, pathogen, w_source))
```

```
# A tibble: 24 x 7
   vol_type  pathogen w_source ill_risk_mean ill_risk_median ill_risk_p5
   <chr>     <chr>    <chr>            <dbl>           <dbl>       <dbl>
 1 EPA_16_70 ETEC     tap          0.0000838   0.00000000559    1.28e-12
 2 EPA_16_70 campy    tap          0.0000868   0.00000000252    5.00e-13
 3 EPA_16_70 rota     tap          0.00684     0.0000000262     2.37e-14
 4 EPA_16_70 ETEC     stored       0.0207      0.000749         4.22e- 9
 5 EPA_16_70 campy    stored       0.0335      0.000355         1.73e- 9
 6 EPA_16_70 rota     stored       0.0688      0.00330          2.54e-10
 7 EPA_All   ETEC     tap          0.0000658   0.00000000441    9.77e-13
 8 EPA_All   campy    tap          0.0000912   0.00000000195    3.64e-13
 9 EPA_All   rota     tap          0.00593     0.0000000162     1.52e-14
10 EPA_All   ETEC     stored       0.0199      0.000631         3.16e- 9
# i 14 more rows
# i 1 more variable: ill_risk_p95 <dbl>
```

Yearly infection risk

First, an overview of the overall illness risk simulations by plotting an histogram

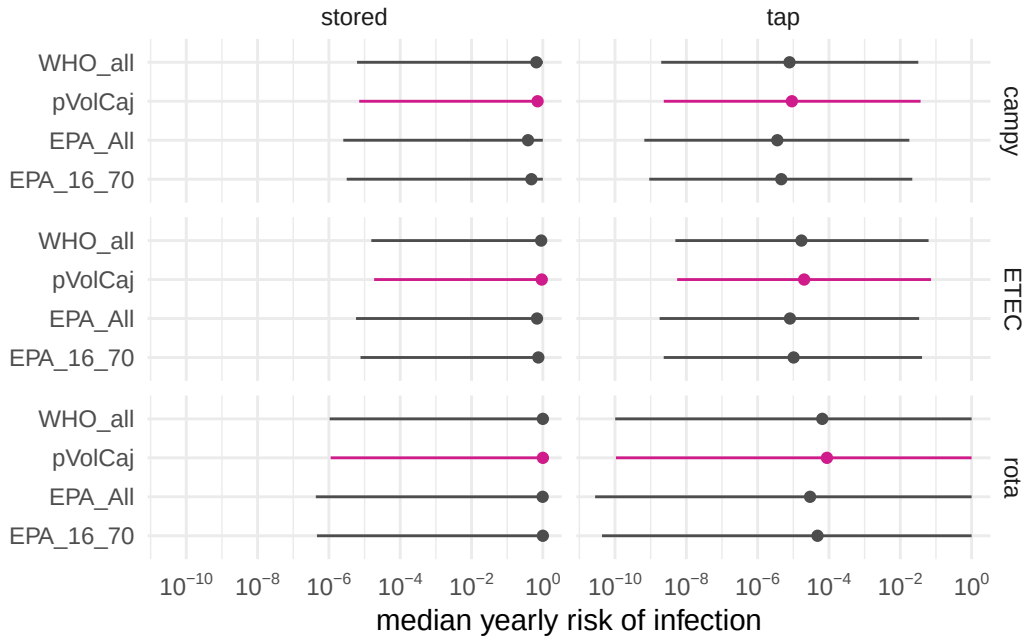
```r
hist(log(df_yearly_risk$yearly_risk), main="simulated yearly infection risks", xlab = "log
```



Through the code below we visualize the results of estimating the yearly risk of infection using the beta-poisson model of the 3 pathogens of study.

```r
risk_year <- df_yearly_risk |> summarise(across(yearly_risk,
                              list(mean = mean,
                                   median = median,
                                   p5 = \(x) quantile(x,0.05),
                                   p95 = \(x) quantile(x,0.95)
                                   )
                              ),
                         .by = c(vol_type, w_source, pathogen)
                         ) |>
  ggplot(aes(x = vol_type))+
  geom_linerange(aes(ymin = yearly_risk_p5,ymax = yearly_risk_p95, col=vol_type))+
  geom_point(aes(y=yearly_risk_median, col=vol_type))+
  facet_grid(pathogen~w_source)+
  coord_flip()+
  scale_y_log10(breaks = scales::trans_breaks
                ("log10", function(x) 10^x),
                labels = scales::trans_format
                ("log10", scales::math_format(10^.x)))+
  scale_color_manual(values=c("gray30", "gray30","#d01c8b","gray30"))+
  theme_minimal()+
```

```
  labs(x = NULL , y = "median yearly risk of infection")+
  theme(legend.position = "none")


print(risk_year)
```



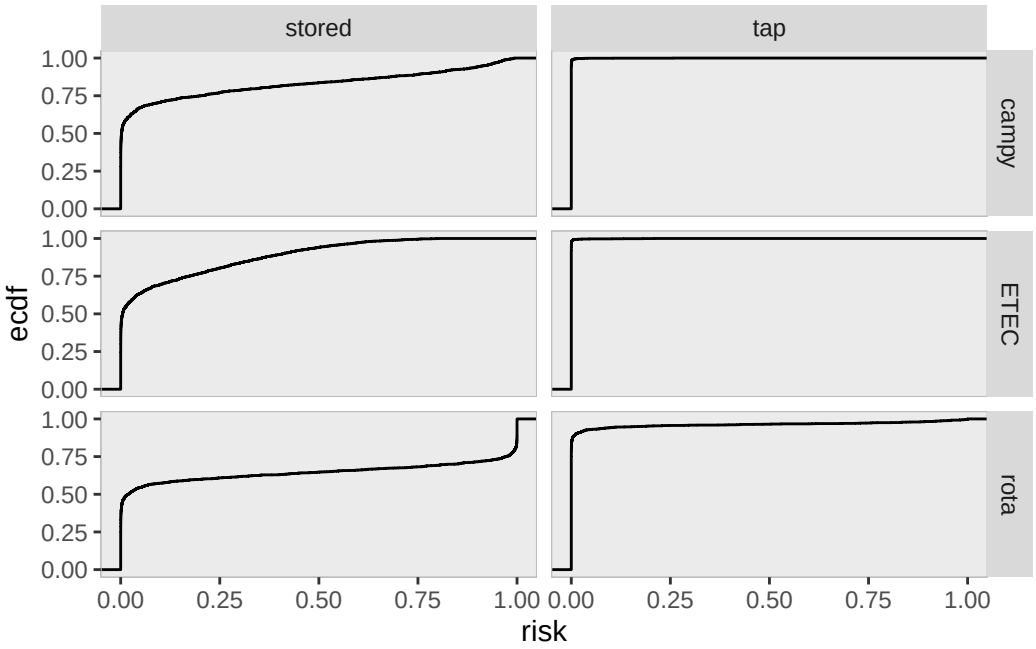

```
df_yearly_risk |> summarise(across(yearly_risk,
                              list(mean = mean,
                                   median = median,
                                   p5 = \(x) quantile(x,0.05),
                                   p95 = \(x) quantile(x,0.95)
                                   )
                              ),
                          .by = c(w_source, vol_type))
```

```
# A tibble: 8 x 6
  w_source vol_type  yearly_risk_mean yearly_risk_median yearly_risk_p5
  <chr>    <chr>                <dbl>              <dbl>          <dbl>
1 tap      EPA_16_70           0.0557          0.0000109        4.39e-10
2 stored   EPA_16_70           0.549           0.793            2.29e- 6
3 tap      EPA_All             0.0532          0.00000807       3.18e-10
4 stored   EPA_All             0.536           0.701            1.94e- 6
5 tap      pVolCaj             0.0615          0.0000208        1.08e- 9
6 stored   pVolCaj             0.580           0.943            5.09e- 6
7 tap      WHO_all             0.0598          0.0000171        9.91e-10
```

```
8 stored   WHO_all              0.572          0.904            4.69e- 6
# i 1 more variable: yearly_risk_p95 <dbl>
```

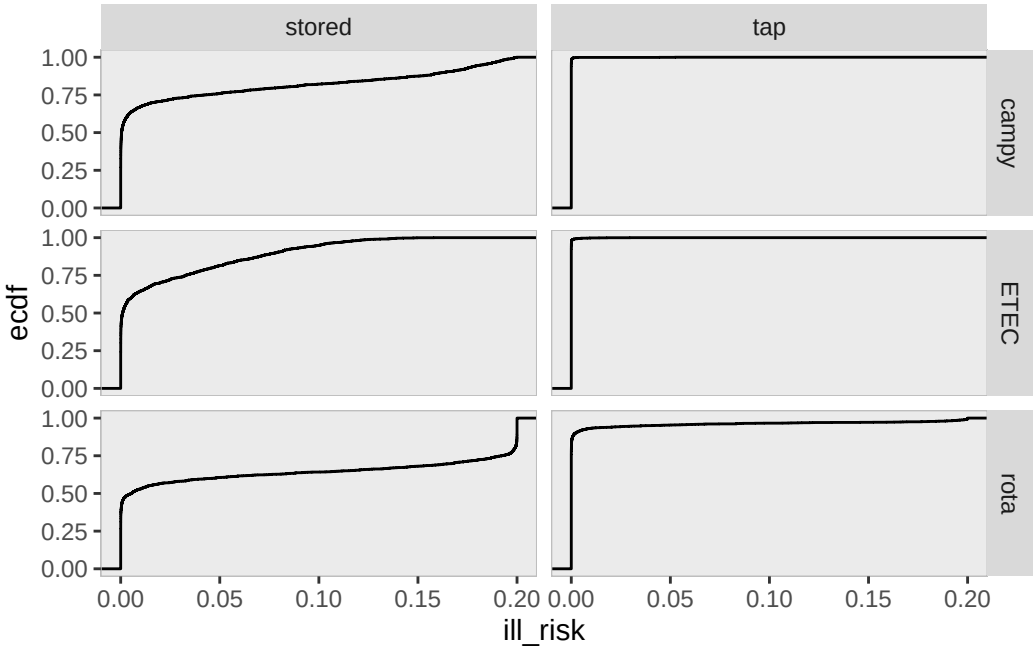
Daily risk CDFs per pathogen

```
df_risk %>%
  sample_n(1e4) %>%
  ggplot(aes(x=risk))+
  stat_ecdf(geom = "step")+
  facet_grid(pathogen~w_source)+
  theme(panel.grid = element_blank(),
        panel.border = element_rect(linewidth = 0.3,colour = "grey",fill = NA))
```



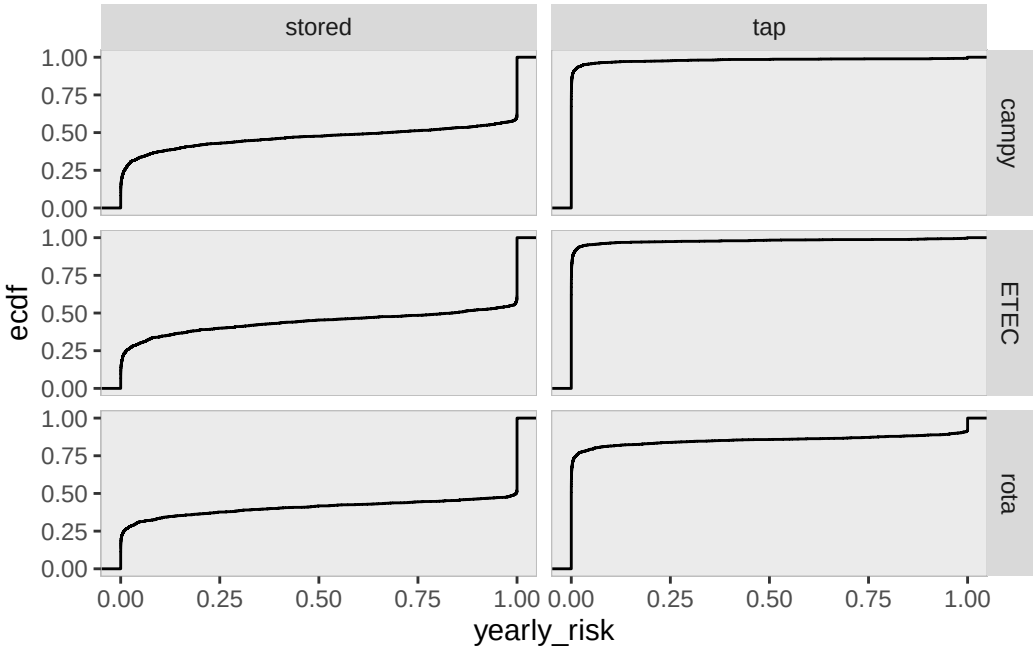
Daily illness risk CDFs per pathogen

```
df_illness_risk %>%
  sample_n(1e4) %>%
  ggplot(aes(x=ill_risk))+
  stat_ecdf(geom = "step")+
  facet_grid(pathogen~w_source)+
  theme(panel.grid = element_blank(),
        panel.border = element_rect(linewidth = 0.3,colour = "grey",fill = NA))
```



Yearly risk of infection CDFs per pathogen

```
df_yearly_risk %>%
  sample_n(1e4) %>%
  ggplot(aes(x=yearly_risk))+
  stat_ecdf(geom = "step")+
  facet_grid(pathogen~w_source)+
  theme(panel.grid = element_blank(),
        panel.border = element_rect(linewidth = 0.3,colour = "grey",fill = NA))
```



Sensitivity analysis

```
cor1 <- df_risk |>
  summarise(cor_Risk_Vol = cor(risk,vol_L, method = "spearman"),
            cor_Risk_Conc_P = cor(risk,conc_p, method = "spearman"),
            cor_Risk_dose = cor(risk,dose, method = "spearman"),
            #cor_Risk_N50 = cor(risk,N50),
            #cor_Risk_a = cor(risk,a),
            .by = c(w_source,pathogen))
```

Correlation plots

risk vs vol

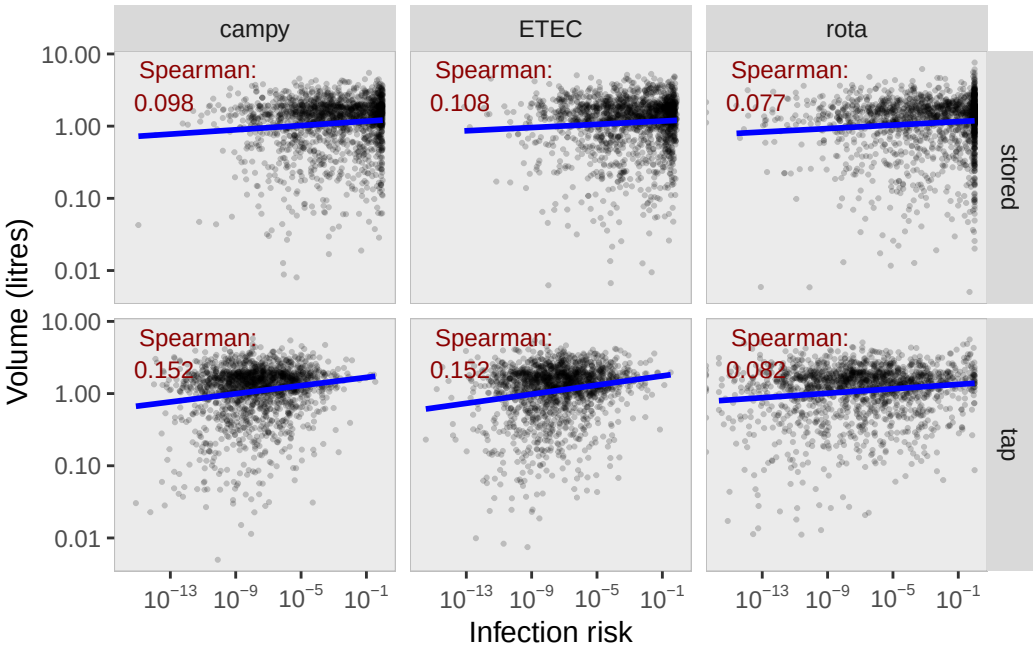
```
corr_riskvsvol <- df_risk %>%
  sample_n(1e4) |>
  ggplot(aes(x=risk, y=vol_L))+
  geom_point(shape = 19,alpha = 0.2,size = 0.4)+
  geom_smooth(method = "lm", se = FALSE, color = "blue")+
  geom_text(data = cor1,
            aes(label = paste("Spearman: \n", round(cor_Risk_Vol, 3))),
            x = -Inf, y = Inf, size = 3,
            hjust = -0.2, vjust = 1.2, color = "darkred") +
```

```
  scale_x_log10(labels = trans_format("log10", math_format(10^.x)))+
  scale_y_log10()+
  # theme_ipsum_rc()+
  labs(y = "Volume (litres)",x = "Infection risk")+
  facet_grid(w_source~pathogen)+
  theme(panel.grid = element_blank(),
        panel.border = element_rect(linewidth = 0.3,colour = "grey",fill = NA)
        )

print(corr_riskvsvol)
```

```
Warning in scale_x_log10(labels = trans_format("log10", math_format(10^.x))): log-10 trans
log-10 transformation introduced infinite values.
```

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

```
Warning: Removed 12 rows containing non-finite outside the scale range
(`stat_smooth()`).
```



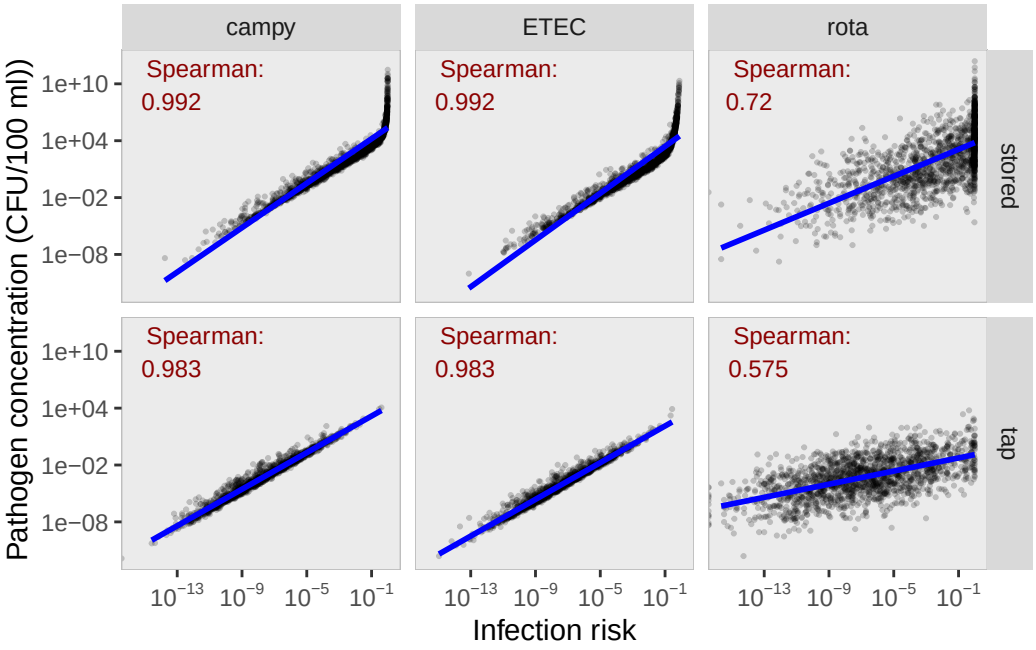
risk vs pathogen concentration

```r
corr_riskvsconcP <- df_risk %>%
  sample_n(1e4) |>
  ggplot(aes(x=risk, y=conc_p))+
  geom_point(shape = 19,alpha = 0.2,size = 0.4)+
  geom_smooth(method = "lm", se = FALSE, color = "blue")+
  geom_text(data = cor1,
            aes(label = paste("Spearman: \n", round(cor_Risk_Conc_P, 3))),
            x = -Inf, y = Inf, size = 3,
            hjust = -0.2, vjust = 1.2, color = "darkred") +
  scale_x_log10(labels = trans_format("log10", math_format(10^.x)))+
  scale_y_log10()+
  # theme_ipsum_rc()+
  labs(y = "Pathogen concentration (CFU/100 ml))",x = "Infection risk")+
  facet_grid(w_source~pathogen)+
  theme(panel.grid = element_blank(),
        panel.border = element_rect(linewidth = 0.3,colour = "grey",fill = NA)
        )
print(corr_riskvsconcP)
```

```
Warning in scale_x_log10(labels = trans_format("log10", math_format(10^.x))): log-10 trans
log-10 transformation introduced infinite values.
```

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

```
Warning: Removed 16 rows containing non-finite outside the scale range
(`stat_smooth()`).
```



risk vs dose

```
df_risk %>%
  sample_n(1e4) |>
  ggplot(aes(x=risk, y=dose))+
  geom_point(shape = 19,alpha = 0.2,size = 0.4)+
  geom_smooth(method = "lm", se = FALSE, color = "blue")+
  geom_text(data = cor1,
            aes(label = paste("Spearman: \n", round(cor_Risk_dose, 3))),
            x = -Inf, y = Inf, size = 3,
            hjust = -0.2, vjust = 1.2, color = "darkred") +
  scale_x_log10(labels = trans_format("log10", math_format(10^.x)))+
  scale_y_log10()+
  # theme_ipsum_rc()+
  labs(y = "Dose",x = "Infection risk")+
  facet_grid(w_source~pathogen)+
  theme(panel.grid = element_blank(),
        panel.border = element_rect(linewidth = 0.3,colour = "grey",fill = NA)
        )
```

```
Warning in scale_x_log10(labels = trans_format("log10", math_format(10^.x))): log-10 trans
log-10 transformation introduced infinite values.
```

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

```
Warning: Removed 9 rows containing non-finite outside the scale range
(`stat_smooth()`).
```