

```
#####  
#-----HEADER-----  
#####  
# This code was written to evaluate the performance of 14 THM models |  
# developed for DWTP, DWDS, and PP systems. It uses data collected from |  
# literature to form representative data sets for DWTP and DWDS systems. |  
# Distributions are fit to the data and MC simulations are run to create |  
# data frames of specified size. The water quality data are used to produce |  
# outputs of the models. Descriptive statistics of the water quality |  
# variables, model outputs, as well as ranking of model outputs compared to |  
# collected THM data are calculated. Graphical representation of water |  
# quality variable data and model outputs are also produced. Care was taken |  
# to include sources and units where necessary. |  
# |  
# Written by Derek Hogue, PhD student at Arizona State University, |  
# School of Sustainable Engineering and the Built Environment, |  
# for 'Predictive capability of THM models for drinking water treatment and |  
# distribution' published in Environmental Science: Water Research and |  
# Technology |  
# Under the guidance of Dr. Treavor Boyer, Associate professor at Arizona |  
# State University, School of Sustainable Engineering and the Built |  
# Environment |  
# Contact: dahogue@asu.edu, thboyer@asu.edu |  
#####
```

```
# Required libraries  
require(fitdistrplus)  
require(LaplacesDemon)  
require(ggplot2)  
require(gridExtra)  
require(dplyr)  
require(scales)  
require(moments)
```

```
##### Functions #####
```

```
round_df <- function(x, digits) {  
  # round all numeric variables  
  # x: data frame  
  # digits: number of digits to round  
  numeric_columns <- sapply(x, mode) == 'numeric'  
  x[numeric_columns] <- round(x[numeric_columns], digits)  
  x  
}
```

```
quant_calc <- function(x, probb) {  
  # calculate quantile for supplied df cols and return as row values  
  # x: data frame  
  # probb: quantile  
  p <- data.frame(matrix(data = NA, nrow = ncol(x)))  
  for (i in 1:nrow(p)) {  
    p[i,] <- round_df(quantile(x[,i], probs = probb), 2)  
  }  
  p  
}
```

```
get_stats <- function(x) {
```

```

# create new df with mean, sd, 5%, and 95% quantile values for supplied df cols
# x: data frame
varstats_df <- data.frame(matrix(data = NA, nrow = ncol(x), ncol = 4))
row.names(varstats_df) <- colnames(x)
varstats_df[,1] <- round_df(sapply(x, mean), 2)
varstats_df[,2] <- round_df(sapply(x, sd), 2)
varstats_df[,3] <- cbind(quant_calc(x, 0.05))
varstats_df[,4] <- cbind(quant_calc(x, 0.95))
colnames(varstats_df) <- c("Mean", "SD", "5%", "95%")
varstats_df
}

```

```

compare_moments_weighted <- function(model_data, thm_data_1, thm_data_2, weights) {
# create 2 df's which 1) report the moments of the passed model data and thm data
# and 2) report the weighted sum of the abs difference between model data moments,
# and THM data moments
# model_data: model data (WTP, DS, or PP)
# thm_data: THM data 1 and 2 (WTP and DS)
# weights: user specified weights for moments

if (length(weights) != 3) {
  stop("Length of weights must be equal to 3")
}

```

```

comparison <- data.frame(matrix(ncol = ncol(model_data) + 2))
colnames(comparison) <- c(colnames(model_data), colnames(thm_data_1),
  colnames(thm_data_2))

```

```

for (i in 1:(ncol(model_data) + 2)) {
  if (i <= ncol(model_data)) {
    comparison[1, i] <- mean(model_data[, i])
    comparison[2, i] <- var(model_data[, i])
    comparison[3, i] <- skewness(model_data[, i])
  } else if (i == (ncol(model_data) + 1)) {
    comparison[1, i] <- mean(thm_data_1)
    comparison[2, i] <- var(thm_data_1)
    comparison[3, i] <- skewness(thm_data_1)
  } else {
    comparison[1, i] <- mean(thm_data_2)
    comparison[2, i] <- var(thm_data_2)
    comparison[3, i] <- skewness(thm_data_2)
  }
}
}

```

```

weighted_difference <- data.frame(matrix(ncol = ncol(model_data), nrow = 1))
colnames(weighted_difference) <- colnames(model_data)

```

```

for (i in 1:ncol(model_data)) {
  diff_1 <- sum(abs(comparison[1:3, i] - comparison[1:3, (ncol(model_data) + 1)]) * weights)
  diff_2 <- sum(abs(comparison[1:3, i] - comparison[1:3, (ncol(model_data) + 2)]) * weights)
  weighted_difference[1, i] <- mean(c(diff_1, diff_2))
}

```

```

comparison <- round_df(comparison, 1)
weighted_difference <- round_df(weighted_difference, 0)

```

```

return(list(comparison = comparison, weighted_difference = weighted_difference))
}

if(.Platform$OS.type=="windows") {
  # change quartz call to windows based on operating system
  quartz<-function() windows()
}

prep_data <- function(variable, fit_variable) {
  # create data frame for ggplot
  data.frame(x = c(density(variable)$x, density(fit_variable)$x),
            y = c(density(variable)$y, density(fit_variable)$y),
            Type = rep(c("Original", "Fitted"), each = length(density(variable)$x)))
}

##### Model variables and associated data #####
# Runs for MC simulation
runs = 1E5

##### DWTP WQVs
# Data and sources reported either as raw data, or n, mean & sd

Cld.wtp <- c(rnorm(88, 2.59, 0.409), rnorm(731, 2.48, 0.36), rnorm(112, 4.51, 1.39))
Clr.wtp <- c(rnorm(573, 1.21, 0.063), rnorm(126, 0.88, 0.076), rnorm(117, 1, 0.2))
Cld.wtp <- Cld.wtp[Cld.wtp > 0 & Cld.wtp < 4]
Clr.wtp <- Clr.wtp[Clr.wtp > 0 & Clr.wtp < 4]
# [mg/L] data from Golfinopoulos et al. 1998, Sohn et al. 2001, Uyak et al. 2005,
# Godo-Pla et al. 2021, Chaves et al. 2021,

Br.wtp <- c(rnorm(76, 0.67, 0.273), rnorm(21, 0.55, 0.066), rnorm(573, 0.36, 0.13))
Br.wtp <- Br.wtp[Br.wtp > 0]
# [mg/L] data from Golfinopoulos et al. 1998, Sohn et al. 2001, Godo-Pla et al. 2021,

pH.wtp <- c(rnorm(88, 7.72, 0.188), rnorm(731, 7.61, 0.089), c(7.42, 7.38, 7.51,
  7.39, 7.35, 7.39, 7.48, 7.42, 7.56, 7.47, 7.52, 7.66, 7.7, 7.42, 7.35,
  7.68, 7.53, 7.48, 7.39, 7.52, 7.44, 7.48), c(8.31, 8.12, 8.11, 8.17,
  8.22, 8.19, 8.09, 8.02, 8.11, 8.08, 8.05, 8.21, 8.25, 8.17, 8.16, 8.18,
  8.19, 8.22, 8.27, 8.25, 8.24, 8.26, 8.15, 8.41, 8.05, 8.08, 8.17, 8.16,
  8.15, 8.24, 8.02, 8), rnorm(52, 7.14, 0.54),
  rnorm(52, 8.3, 0.77), rnorm(52, 7.75, 0.32), rnorm(52, 7.24, 0.25),
  rnorm(573, 7.45, 0.19))
pH.wtp <- pH.wtp[pH.wtp > 6.5 & pH.wtp < 8.5]
# data from Golfinopoulos et al. 1998, Sohn et al. 2001, Golfinopoulos et al. 2002,
# Rodriguez et al. 2003, Godo-Pla et al. 2021,

Temp.wtp <- c(rnorm(88, 16.11, 4.082), rnorm(731, 13.6, 3.55), c(22.75, 21.63,
  21, 20.25, 19.75, 19.75, 19.5, 18.75, 18.5, 18.25, 16.75, 16.28,
  15, 14, 13.5, 12.38, 12.33, 11, 10.68, 10, 10), c(22.5, 22.25,
  18.5, 18.5, 17.5, 17.5, 17.1, 17, 16.75, 16.15, 16, 15.25, 15,
  15, 15, 14.5, 14.3, 14.2, 14, 13.25, 13.1, 13, 13, 11.5, 11, 10,
  9.8, 9, 9, 8.5, 8.5, 8), rnorm(124, 15.23, 5.68), rnorm(573, 17.45, 4.9))
Temp.wtp <- Temp.wtp[Temp.wtp > 0]
# [°C] data from Golfinopoulos et al. 1998, Sohn et al. 2001, Golfinopoulos et al. 2002,
# Uyak et al. 2005, Godo-Pla et al. 2021,

```

```

DOC.wtp <- c(rnorm(731, 1.8, 0.23))
DOC.wtp <- DOC.wtp[DOC.wtp > 0]
# [mg/L] data from Sohn et al. 2001,

TOC.wtp <- c(rnorm(52, 2.24, 0.51), rnorm(52, 2.27, 0.77),
             rnorm(52, 3.95, 0.65), rnorm(52, 2.15, 0.22), rnorm(52, 2.34, 0.35),
             rnorm(573, 1.21, 0.34), rnorm(120, 1.05, 0.14), rnorm(108, 2.69, 0.55))
TOC.wtp <- TOC.wtp[TOC.wtp > 0]
# [mg/L] data from Rodriguez et al. 2003, Godo-Pla et al. 2021,
# Chaves et al. 2021

RT.wtp <- c(rnorm(573, 11.6, 21.2))
RT.wtp <- RT.wtp[RT.wtp > 0]
# [hours] data from Godo-Pla et al. 2021,

UVA.wtp <- c(rnorm(52, 0.033, 0.017), rnorm(52, 0.087, 0.029),
             rnorm(52, 0.123, 0.035), rnorm(52, 0.042, 0.018), rnorm(52, 0.053, 0.025),
             rnorm(26, 0.02, 0.007), rnorm(22, 0.023, 0.008), rnorm(17, 0.031, 0.02),
             rnorm(16, 0.024, 0.019), rnorm(573, 0.0203, 0.0056))
UVA.wtp <- UVA.wtp[UVA.wtp > 0 & UVA.wtp < 0.3]
# [1/cm] from Rodriguez et al. 2003, Chang et al. 2010,
# Godo-Pla et al. 2021,

Alk.wtp <- c(rnorm(116, 128, 23.34))
Alk.wtp <- Alk.wtp[Alk.wtp > 0]
# [mg/L] data from Uyak et al. 2005,

THM.wtp <- c(rnorm(88, 47.54, 17.982), rnorm(96, 68.7, 18), rnorm(14, 112.31, 19.74),
             rnorm(14, 58.53, 16.27), rnorm(14, 40.94, 12.1), rnorm(14, 41.78, 12.6),
             rnorm(52, 34.6, 12), rnorm(45, 35.1, 8.4), rnorm(40, 34, 12.9),
             rnorm(47, 35.3, 11.5), rnorm(573, 19.52, 20), rnorm(124, 20.4, 9.3),
             rnorm(109, 37.9, 15.7))
THM.wtp <- THM.wtp[THM.wtp > 0]
# [ug/L] data from Golfinopoulos et al. 1998, Uyak et al. 2005,
# Dominguez-Tello et al. 2015, Dominguez-Tello et al. 2017, Godo-Pla et al. 2021,
# Chaves et al. 2021

##### DWDS WQVs

Cl.ds <- c(rnorm(55, 0.15, 0.13), rnorm(94, 0.09, 0.09), rnorm(55, 0.27, 0.24),
          rnorm(35, 0.34, 0.12), rnorm(342, 0.61, 0.29), rnorm(112, 0.37, 0.38),
          rnorm(209, 0.79, 0.33), rnorm(230, 0.42, 0.45), rnorm(573, 1.21, 0.063),
          rnorm(216, 0.45, 0.37), rnorm(74, 1.92, 0.6), rnorm(117, 1.1, 0.3),
          rnorm(624, 0.38, 0.37), rnorm(72, 0.5, 0.02))
Cl.ds <- Cl.ds[Cl.ds > 0 & Cl.ds < 2]
# [mg/L] data from Mouly et al. 2010, Tsitsifli and Kanakoudis 2020,
# Osorio et al. 2010, Godo-Pla et al. 2021, Kelly-Coto et al. 2022, Abdullah et al. 2003,
# Feungpean et al. 2015, Zhang et al. 2015,

Br.ds <- c(rnorm(70, 0.216, 0.163), rnorm(106, 0.226, 0.148), rnorm(66, 0.144, 0.121),
          rnorm(573, 0.36, 0.13), rnorm(72, 0.01, 0.001))
Br.ds <- Br.ds[Br.ds > 0 & Br.ds < 1]
# [mg/L] data from Mouly et al. 2010, Godo-Pla et al. 2021, Zhang et al. 2015,

```

```

pH.ds <- c(rnorm(70, 7.8, 0.2), rnorm(106, 8, 0.2), rnorm(66, 8.2, 0.2),
          rnorm(35, 7.93, 0.33), rnorm(7.45, 0.19), rnorm(216, 7.3, 1.01),
          rnorm(74, 7.2, 0.6), rnorm(117, 7.4, 0.26), rnorm(30.18, 2.76),
          rnorm(72, 7.4, 0.03))
pH.ds <- pH.ds[pH.ds > 6.5 & pH.ds < 8.5]
# data from Mouly et al. 2010, Tsitsifli and Kanakoudis 2020, Godo-Pla et al. 2021,
# Kelly-Coto et al. 2022, Abdullah et al. 2003, Feungpean et al. 2015, Zhang et al. 2015,

Temp.ds <- c(rnorm(70, 13.8, 4.1), rnorm(106, 15, 5), rnorm(66, 12.8, 4.8),
            rnorm(573, 17.45, 4.9), rnorm(216, 22, 5.7), rnorm(74, 29.1, 1.0),
            rnorm(117, 29.6, 0.8), rnorm(72, 13.1, 0.4))
Temp.ds <- Temp.ds[Temp.ds > 0]
# [°C] data from Mouly et al. 2010, Godo-Pla et al. 2021, Kelly-Coto et al. 2022,
# Abdullah et al. 2003, Zhang et al. 2015,

DOC.ds <- c(rnorm(216, 0.48, 0.37))
DOC.ds <- DOC.ds[DOC.ds > 0]
# [mg/L] data from Kelly-Coto et al. 2022,

TOC.ds <- c(rnorm(70, 2.3, 0.8), rnorm(106, 2.7, 0.7), rnorm(66, 2, 0.4),
            rnorm(35, 5.01, 9.94), rnorm(342, 2.1, 0.66), rnorm(112, 3.88, 2.12),
            rnorm(209, 1.41, 0.47), rnorm(230, 0.93, 0.85), rnorm(573, 1.21, 0.81),
            rnorm(194, 1.03, 0.26), rnorm(216, 0.5, 0.38), rnorm(74, 2.16, 0.6),
            rnorm(117, 2.22, 0.36), rnorm(624, 3.09, 1.27), rnorm(72, 3.7, 0.3))
TOC.ds <- TOC.ds[TOC.ds > 0 & TOC.ds < 5]
# [mg/L] data from Mouly et al. 2010, Tsitsifli and Kanakoudis 2020, Osorio et al. 2010,
# Godo-Pla et al. 2021, Chaves et al. 2021, Kelly-Coto et al. 2022, Abdullah et al. 2003,
# Feungpean et al. 2015, Zhang et al. 2015,

RT.ds <- c(rnorm(573, 11.6, 21.2), rnorm(68, 19.5, 2.8), rnorm(120, 63.1, 32.3),
            rnorm(96, 94.8, 38.4), rnorm(72, 12.4, 1.9))
RT.ds <- RT.ds[RT.ds > 1]
# [hours] data from Godo-Pla et al. 2021, Mouly et al. 2010, Zhang et al. 2015,

UVA.ds <- c(rnorm(70, 0.03, 0.02), rnorm(106, 0.04, 0.02), rnorm(66, 0.03, 0.01),
            rnorm(573, 0.0203, 0.0056), rnorm(216, 0.0082, 0.0093), rnorm(72, 0.03, 0.001))
UVA.ds <- UVA.ds[UVA.ds > 0 & UVA.ds < 0.1]
# [1/cm] from Mouly et al. 2010, Godo-Pla et al. 2021, Kelly-Coto et al. 2022,
# Zhang et al. 2015,

cond.ds <- c(rnorm(35, 690, 171.7), rnorm(342, 1172.76, 615.89), rnorm(112, 373.4, 171.56),
            rnorm(209, 1052.83, 261.56), rnorm(230, 781.76, 496.93))
cond.ds <- cond.ds[cond.ds > 100 & cond.ds < 2000]
# [uS/cm] data from Tsitsifli and Kanakoudis 2020, Osorio et al. 2010,

bicarb.ds <- c(rnorm(342, 229.94, 76.4), rnorm(112, 134.8, 79.12), rnorm(209, 144.27, 36.79),
              rnorm(230, 254.53, 68.03))
bicarb.ds <- bicarb.ds[bicarb.ds > 0]
# [mg/L] data from Osorio et al. 2010,

THM.ds <- c(rnorm(45, 38.2, 7.5), rnorm(45, 44.1, 11.6), rnorm(33, 45.6, 16.4),
            rnorm(40, 39.9, 13.8), rnorm(35, 10.7, 14.9), rnorm(56, 115.21, 17.44),
            rnorm(56, 81.99, 16.38), rnorm(56, 64.42, 11.28), rnorm(56, 48.8, 18.05),
            rnorm(573, 19.52, 20), rnorm(746, 30.5, 10), rnorm(216, 10.64, 15.24),
            rnorm(74, 42.8, 28.9), rnorm(117, 69.9, 22.2), rnorm(66.12, 27.38),
            rnorm(72, 30.3, 1.1))

```

```

THM.ds <- THM.ds[THM.ds > 0]
# [ug/L] data from Dominguez-Tello et al. 2017, Tsitsifli and Kanakoudis 2020,
# Dominguez-Tello et al. 2015, Godo-Pla et al. 2021, Kelly-Coto et al. 2022,
# Abdullah et al. 2003, Feungpean et al. 2015, Zhang et al. 2015,

##### Variable distributions #####
# descdist() used to find best distributions based on available data
# All WQVs: Clr, Cld, Br, pH, Temp, DOC, TOC, RT, UVA, Alk, cond, bicarb, THM
# rtrunc() used to produce fit distributions based on parameters and based on
# truncated data to prevent impossible/unlikely outliers for better comparison

# descdist(Cld.wtp)
Cld.wtp_est <- fitdist(Cld.wtp, "norm", method = "mle")
Cld.wtp_fit <- rtrunc(runs, "norm", a = 0, b = 4, mean = Cld.wtp_est$estimate[1],
                    sd = Cld.wtp_est$estimate[2])
# plot(density(Cld.wtp))
# lines(density(Cld.wtp_fit), col = "red")

# descdist(Clr.wtp)
Clr.wtp_est <- fitdist(Clr.wtp, "logis", method = "mle")
Clr.wtp_fit <- rtrunc(runs, "logis", a = 0, b = 2, location = Clr.wtp_est$estimate[1],
                    scale = Clr.wtp_est$estimate[2])
# plot(density(Clr.wtp))
# lines(density(Clr.wtp_fit), col = "red")

# descdist(Cl.ds)
Cl.ds_est <- fitdist(Cl.ds, "logis", method = "mle")
Cl.ds_fit <- rtrunc(runs, "logis", a = 0, b = 2, location = Cl.ds_est$estimate[1],
                  scale = Cl.ds_est$estimate[2])
# plot(density(Cl.ds))
# lines(density(Cl.ds_fit), col = "red")

# descdist(Br.wtp)
Br.wtp_est <- fitdist(Br.wtp, "lnorm", method = "mle")
Br.wtp_fit <- rtrunc(runs, "lnorm", a = 0, b = 1, meanlog = Br.wtp_est$estimate[1],
                  sdlog = Br.wtp_est$estimate[2])
# plot(density(Br.wtp))
# lines(density(Br.wtp_fit), col = "red")

# descdist(Br.ds)
Br.ds_est <- fitdist(Br.ds, "norm", method = "mle")
Br.ds_fit <- rtrunc(runs, "norm", a = 0, b = 1, mean = Br.ds_est$estimate[1],
                  sd = Br.ds_est$estimate[2])
# plot(density(Br.ds))
# lines(density(Br.ds_fit), col = "red")

# descdist(pH.wtp)
pH.wtp_est <- fitdist(pH.wtp, "logis", method = "mle")
pH.wtp_fit <- rtrunc(runs, "logis", a = 6.5, b = 8.5, location = pH.wtp_est$estimate[1],
                  scale = pH.wtp_est$estimate[2])
# plot(density(pH.wtp))
# lines(density(pH.wtp_fit), col = "red")

# descdist(pH.ds)

```

```

pH.ds_est <- fitdist(pH.ds, "norm", method = "mle")
pH.ds_fit <- rtrunc(runs, "norm", a = 6.5, b = 8.5, mean = pH.ds_est$estimate[1],
                  sd = pH.ds_est$estimate[2])
# plot(density(pH.ds))
# lines(density(pH.ds_fit), col = "red")

# descdist(Temp.wtp)
Temp.wtp_est <- fitdist(Temp.wtp, "norm", method = "mle")
Temp.wtp_fit <- rtrunc(runs, "norm", a = 1, b = 30, mean = Temp.wtp_est$estimate[1],
                    sd = Temp.wtp_est$estimate[2])
# plot(density(Temp.wtp))
# lines(density(Temp.wtp_fit), col = "red")

# descdist(Temp.ds)
Temp.ds_est <- fitdist(Temp.ds, "norm", method = "mle")
Temp.ds_fit <- rtrunc(runs, "norm", a = 1, b = 35, mean = Temp.ds_est$estimate[1],
                    sd = Temp.ds_est$estimate[2])
# plot(density(Temp.ds))
# lines(density(Temp.ds_fit), col = "red")

# descdist(DOC.wtp)
DOC.wtp_est <- fitdist(DOC.wtp, "norm", method = "mle")
DOC.wtp_fit <- rtrunc(runs, "norm", a = 1, b = 3, mean = DOC.wtp_est$estimate[1],
                    sd = DOC.wtp_est$estimate[2])
# plot(density(DOC.wtp))
# lines(density(DOC.wtp_fit), col = "red")

# descdist(DOC.ds)
DOC.ds_est <- fitdist(DOC.ds, "norm", method = "mle")
DOC.ds_fit <- rtrunc(runs, "norm", a = 0, b = 2, mean = DOC.ds_est$estimate[1],
                    sd = DOC.ds_est$estimate[2])
# plot(density(DOC.ds))
# lines(density(DOC.ds_fit), col = "red")

# descdist(TOC.wtp)
TOC.wtp_est <- fitdist(TOC.wtp, "gamma", method = "mle")
TOC.wtp_fit <- rtrunc(runs, "gamma", a = 0.5, b = 6, shape = TOC.wtp_est$estimate[1],
                    rate = TOC.wtp_est$estimate[2])
# plot(density(TOC.wtp))
# lines(density(TOC.wtp_fit), col = "red")

# descdist(TOC.ds)
TOC.ds_est <- fitdist(TOC.ds, "norm", method = "mle")
TOC.ds_fit <- rtrunc(runs, "norm", a = 0, b = 5, mean = TOC.ds_est$estimate[1],
                    sd = TOC.ds_est$estimate[2])
# plot(density(TOC.ds))
# lines(density(TOC.ds_fit), col = "red")

# descdist(RT.wtp)
RT.wtp_est <- fitdist(RT.wtp, "gamma", method = "mle")
RT.wtp_fit <- rtrunc(runs, "gamma", a = 0, b = 80, shape = RT.wtp_est$estimate[1],
                    rate = RT.wtp_est$estimate[2])
# plot(density(RT.wtp))
# lines(density(RT.wtp_fit), col = "red")

# descdist(RT.ds)

```

```

RT.ds_est <- fitdist(RT.ds, "gamma", method = "mle")
RT.ds_fit <- rtrunc(runs, "gamma", a = 1, b = 200, shape = RT.ds_est$estimate[1],
  rate = RT.ds_est$estimate[2])
# plot(density(RT.ds))
# lines(density(RT.ds_fit), col = "red")

# descdist(UVA.wtp)
UVA.wtp_est <- fitdist(UVA.wtp, "lnorm", method = "mle")
UVA.wtp_fit <- rtrunc(runs, "lnorm", a = 0, b = 0.3, meanlog = UVA.wtp_est$estimate[1],
  sdlog = UVA.wtp_est$estimate[2])
# plot(density(UVA.wtp))
# lines(density(UVA.wtp_fit), col = "red")

# descdist(UVA.ds)
UVA.ds_est <- fitdist(UVA.ds, "lnorm", method = "mle")
UVA.ds_fit <- rtrunc(runs, "lnorm", a = 0, b = 0.1, meanlog = UVA.ds_est$estimate[1],
  sdlog = UVA.ds_est$estimate[2])
# plot(density(UVA.ds))
# lines(density(UVA.ds_fit), col = "red")

# descdist(Alk.wtp)
Alk.wtp_est <- fitdist(Alk.wtp, "norm", method = "mle")
Alk.wtp_fit <- rtrunc(runs, "norm", a = 50, b = 200, mean = Alk.wtp_est$estimate[1],
  sd = Alk.wtp_est$estimate[2])
# plot(density(Alk.wtp))
# lines(density(Alk.wtp_fit), col = "red")

# descdist(cond.ds)
cond.ds_est <- fitdist(cond.ds, "norm", method = "mle")
cond.ds_fit <- rtrunc(runs, "norm", a = 100, b = 2000, mean = cond.ds_est$estimate[1],
  sd = cond.ds_est$estimate[2])
# plot(density(cond.ds))
# lines(density(cond.ds_fit), col = "red")

# descdist(bicarb.ds)
bicarb.ds_est <- fitdist(bicarb.ds, "norm", method = "mle")
bicarb.ds_fit <- rtrunc(runs, "norm", a = 0, b = 500, mean = bicarb.ds_est$estimate[1],
  sd = bicarb.ds_est$estimate[2])
# plot(density(bicarb.ds))
# lines(density(bicarb.ds_fit), col = "red")

##### k.ds

# descdist(THM.wtp)
THM.wtp_est <- fitdist(THM.wtp, "gamma", method = "mle")
THM.wtp_fit <- rtrunc(runs, "gamma", a = 5, b = 150, shape = THM.wtp_est$estimate[1],
  rate = THM.wtp_est$estimate[2])
# plot(density(THM.wtp))
# lines(density(THM.wtp_fit), col = "red")

# descdist(THM.ds)
THM.ds_est <- fitdist(THM.ds, "gamma", method = "mle")
THM.ds_fit <- rtrunc(runs, "gamma", a = 5, b = 150, shape = THM.ds_est$estimate[1],
  rate = THM.ds_est$estimate[2])
# plot(density(THM.ds))
# lines(density(THM.ds_fit), col = "red")

```



```

# Variables and titles for the two sets of plots
variables_wtp <- list(Cld.wtp, Clr.wtp, Br.wtp, pH.wtp, Temp.wtp, DOC.wtp, TOC.wtp,
  RT.wtp, UVA.wtp, Alk.wtp, THM.wtp)
variables_ds <- list(cond.ds, Cl.ds, Br.ds, pH.ds, Temp.ds, DOC.ds, TOC.ds, RT.ds,
  UVA.ds, bicarb.ds, THM.ds)
fit_variables_wtp <- list(Cld.wtp_fit, Clr.wtp_fit, Br.wtp_fit, pH.wtp_fit, Temp.wtp_fit,
  DOC.wtp_fit, TOC.wtp_fit, RT.wtp_fit, UVA.wtp_fit, Alk.wtp_fit, THM.wtp_fit)
fit_variables_ds <- list(cond.ds_fit, Cl.ds_fit, Br.ds_fit, pH.ds_fit, Temp.ds_fit,
  DOC.ds_fit, TOC.ds_fit, RT.ds_fit, UVA.ds_fit, bicarb.ds_fit, THM.ds_fit)
titles_wtp <- c("Cl dose", "Cl residual", "Br", "pH", "Temp", "DOC", "TOC", "RT",
  "UV absorbance", "Alk", "THM DWTP")
titles_ds <- c("Conductivity", "Cl residual", "Br", "pH", "Temp", "DOC", "TOC",
  "RT", "UV absorbance", "Bicarbonate", "THM DWDS")
x_labels_wtp <- c("mg/L", "mg/L", "mg/L", "", "°C", "mg/L", "mg/L", "hours", "1/cm",
  "mg/L-CaCO3", "µg/L")
x_labels_ds <- c("µS/cm", "mg/L", "mg/L", "", "°C", "mg/L", "mg/L", "hours", "1/cm",
  "mg/L-CaCO3", "µg/L")

# Create plots for water treatment plant data
plots_wtp <- lapply(1:length(variables_wtp), function(i) {
  ggplot(prepare_data(variables_wtp[[i]], fit_variables_wtp[[i]]), aes(x = x, y = y, linetype = Type)) +
    geom_line() +
    labs(title = titles_wtp[i], x = x_labels_wtp[i], y = "") +
    theme_minimal() +
    theme(legend.position = "none")
})

# Create plots for distribution system data
plots_ds <- lapply(1:length(variables_ds), function(i) {
  ggplot(prepare_data(variables_ds[[i]], fit_variables_ds[[i]]), aes(x = x, y = y, linetype = Type)) +
    geom_line() +
    labs(title = titles_ds[i], x = x_labels_ds[i], y = "") +
    theme_minimal() +
    theme(legend.position = "none")
})

# Combine and display the plots
quartz()
grid.arrange(grobs = plots_wtp, ncol = 4)
quartz()
grid.arrange(grobs = plots_ds, ncol = 4)

##### WQV Statistics from fitted distributions #####
# Variable statistics for DWTP
DWTP.var <- data.frame(fit_variables_wtp) # Convert DWTP WQV MC data to df
DWTP.varstats <- get_stats(DWTP.var) # Call get stats function
rownames(DWTP.varstats) <- titles_wtp # Assign correct WQV names

# Variable statistics for DWDS
DWDS.var <- data.frame(fit_variables_ds) # Convert DWDS WQV MC data to df
DWDS.varstats <- get_stats(DWDS.var) # Call get stats function
rownames(DWDS.varstats) <- titles_ds # Assign correct WQV names

```

```

##### Model evaluation #####
i <- 0 # Iterate loop once for each data set
for (i in i:1) {
  if (i == 0)
  {
    Clr <- Clr.wtp_fit
    Cld <- Cld.wtp_fit
    Cl <- Clr.wtp_fit
    Br <- Br.wtp_fit
    pH <- pH.wtp_fit
    Temp <- Temp.wtp_fit
    DOC <- DOC.wtp_fit
    TOC <- TOC.wtp_fit
    RT <- RT.wtp_fit
    UVA <- UVA.wtp_fit
    Alk <- Alk.wtp_fit
    cond <- cond.ds_fit
    bicarb <- bicarb.ds_fit
    THM <- THM.wtp_fit

    DWTP.modeldata_TP <- data.frame(matrix(data = NA, nrow = runs, ncol = 8))
    colnames(DWTP.modeldata_TP) <- c("Model_1", "Model_2", "Model_3", "Model_4", "Model_5",
                                     "Model_6", "Model_7", "Model_8")
    DWTP.modeldata_TP[,1] <- (10 ^ -2.534)*((1000*Br) ^ 0.212)*(RT ^ 0.305)*
      (DOC ^ 0.369)*(Temp ^ 0.662)*((Clr/DOC) ^ 0.4)*(pH ^ 2.364)
    # Hong et al. 2016
    DWTP.modeldata_TP[,2] <- -150.833+(pH*40.948)+(Temp*6.153)-
      (Clr*13.876)+(RT*8.1)+(TOC*6.221)+(UVA*292.308)
    DWTP.modeldata_TP[,3] <- 33.436*(pH ^ 0.062)*(Temp ^ 0.069)*
      (Clr ^ -0.048)*(RT ^ 0.018)*(TOC ^ 0.079)*(UVA ^ 0.045)
    # Kumari and Gupta 2015, linear and non-linear
    DWTP.modeldata_TP[,4] <- (10 ^ 1.2146)*(Clr ^ 0.3897)*(RT ^ 0.3142)*(UVA ^ 0.1381)
    # Roth and Cornwell 2018
    DWTP.modeldata_TP[,5] <- 85.928+((( -5.2*(10^-4))*UVA*1000*DOC*log(Cld*1000))^2)+
      ((-6.2*(10^-2))*((Br*1000)+2))+((1.66 *(10^-5))*((Clr*1000)^2))+
      ((3.87*(10^-6))*(((Cld/2)*1000)^2))+(-10.25*pH)+(((7*(10^-3))*((Temp)^2)))+
      ((8.42*(10^-5))*UVA*(Temp^2)*RT*1000*Cld)
    # Shahi et al. 2020
    DWTP.modeldata_TP[,6] <- 6.18*((UVA+1)^3.64)*(TOC^0.462)*(Cld^0.42)*
      ((Br+1)^0.471)*(Temp^0.169)*(pH^0.048)*(RT^0.298)
    # Godo-Pla et al. 2021
    DWTP.modeldata_TP[,7] <- 165-(21.3*pH)+(0.232*Br)+(5.84*Cld*RT*Temp*UVA)
    # Dominguez-Tello et al. 2017
    DWTP.modeldata_TP[,8] <- 1147*(UVA^0.83)*(((Br+1))^0.27)
    # Chen and Westerhoff 2010

    DWDS.modeldata_TP <- data.frame(matrix(data = NA, nrow = runs, ncol = 5))
    colnames(DWDS.modeldata_TP) <- c("Model_9", "Model_10", "Model_11", "Model_12", "Model_13")
    DWDS.modeldata_TP[,1] <- 0.035*(TOC^1.098)*(Cl^0.152)*(Temp^0.609)*
      (pH^1.601)*(RT^0.263)
    # Wert et al. 2012
    m2.c <- (7.5*10^7)*((0.7*TOC)-(2.2*Cld))*exp(-6500/Temp)
    DWDS.modeldata_TP[,2] <- (((11.1*TOC)+20.06)-(((11.1*TOC)+20.06)-(THM.wtp_fit))*
      (exp(((Cld*2.86/24)/exp(m2.c)))*(exp(-m2.c*RT))-1)))
  }
}

```

```

# Cong et al. 2012
DWDS.modeldata_TP[,3] <- (-28.826+(1.583*TOC)+(2.713*log(cond))-(1.307*log(bicarb))+
(3.744*Cl)+(2.427*pH)+(0.102*Temp))^2
# Osorio et al. 2011
DWDS.modeldata_TP[,4] <- 10^(-3.84+(0.633*pH)-(0.1056*(TOC^2)))
# Tsitsifili and Kanakoudis 2020
DWDS.modeldata_TP[,5] <- 14.9+(1.01*THM)+(0.2*pH)-(0.104*Cl*Temp*UVA)
# Dominguez-Tello et al. 2017

PP.modeldata_TP <- data.frame(matrix(data = NA, nrow = runs, ncol = 1))
colnames(PP.modeldata_TP) <- c("Model_1")
PP.modeldata_TP[,1] <- 21.4+(36.9*Cl)+(0.986*THM)+(0.59*TOC)+
(-1.83*Temp)+(-1.21*(TOC-4.1)*(Temp-18.7))
# Salehi et al. 2011
}
if(i == 1) {
Cl <- Cl.ds_fit
Br <- Br.ds_fit
pH <- pH.ds_fit
Temp <- Temp.ds_fit
DOC <- DOC.ds_fit
TOC <- TOC.ds_fit
RT <- RT.ds_fit
UVA <- UVA.ds_fit
Alk <- Alk.wtp_fit
cond <- cond.ds_fit
bicarb <- bicarb.ds_fit
THM <- THM.ds_fit

DWTP.modeldata_DS <- data.frame(matrix(data = NA, nrow = runs, ncol = 8))
colnames(DWTP.modeldata_DS) <- c("Model_1", "Model_2", "Model_3", "Model_4", "Model_5",
"Model_6", "Model_7", "Model_8")
DWTP.modeldata_DS[,1] <- (10 ^ -2.534)*((1000*Br) ^ 0.212)*(RT ^ 0.305)*
(DOC ^ 0.369)*(Temp ^ 0.662)*((Clr/DOC) ^ 0.4)*(pH ^ 2.364)
# Hong et al. 2016
DWTP.modeldata_DS[,2] <- -150.833+(pH*40.948)+(Temp*6.153)-
(Clr*13.876)+(RT*8.1)+(TOC*6.221)+(UVA*292.308)
DWTP.modeldata_DS[,3] <- 33.436*(pH ^ 0.062)*(Temp ^ 0.069)*
(Clr ^ -0.048)*(RT ^ 0.018)*(TOC ^ 0.079)*(UVA ^ 0.045)
# Kumari and Gupta 2015, linear and non-linear
DWTP.modeldata_DS[,4] <- (10 ^ 1.2146)*(Clr ^ 0.3897)*(RT ^ 0.3142)*(UVA ^ 0.1381)
# Roth and Cornwell 2018
DWTP.modeldata_DS[,5] <- 85.928+((-5.2*(10^-4))*UVA*1000*DOC*log(Cld*1000))^2+
((-6.2*(10^-2))*((Br*1000)+2))+((1.66 *(10^-5))*((Clr*1000)^2))+
((3.87*(10^-6))*(((Cld/2)*1000)^2))+(-10.25*pH)+(((7*(10^-3))*((Temp)^2)))+
((8.42*(10^-5))*(UVA*(Temp^2)*RT*1000*Cld))
# Shahi et al. 2020
DWTP.modeldata_DS[,6] <- 6.18*((UVA+1)^3.64)*(TOC^0.462)*(Cld^0.42)*
((Br+1)^0.471)*(Temp^0.169)*(pH^0.048)*(RT^0.298)
# Godo-Pla et al. 2021
DWTP.modeldata_DS[,7] <- 165-(21.3*pH)+(0.232*Br)+(5.84*Cld*RT*Temp*UVA)
# Dominguez-Tello et al. 2017
DWTP.modeldata_DS[,8] <- 1147*(UVA^0.83)*(((Br+1))^0.27)
# Chen and Westerhoff 2010

DWDS.modeldata_DS <- data.frame(matrix(data = NA, nrow = runs, ncol = 5))

```

```

colnames(DWDS.modeldata_DS) <- c("Model_9", "Model_10", "Model_11", "Model_12", "Model_13")
DWDS.modeldata_DS[,1] <- 0.035*(TOC^1.098)*(Cl^0.152)*(Temp^0.609)*
  (pH^1.601)*(RT^0.263)
# Wert et al. 2012
m2.c <- (7.5*10^7)*((0.7*TOC)-(2.2*Cl.d.wtp_fit))*exp(-6500/Temp)
DWDS.modeldata_DS[,2] <- ((11.1*TOC)+20.06)-((((11.1*TOC)+20.06)-(THM.wtp_fit))*
  (exp(((Cl.d.wtp_fit*2.86/24)/exp(m2.c)))*((exp(-m2.c*RT))-1))))
# Cong et al. 2012
DWDS.modeldata_DS[,3] <- (-28.826+(1.583*TOC)+(2.713*log(cond))-(1.307*log(bicarb))+
  (3.744*Cl)+(2.427*pH)+(0.102*Temp))^2
# Osorio et al. 2011
DWDS.modeldata_DS[,4] <- 10^(-3.84+(0.633*pH)-(0.1056*(TOC^-2)))
# Tsitsifili and Kanakoudis 2020
DWDS.modeldata_DS[,5] <- 14.9+(1.01*THM.wtp_fit)+(0.2*pH)-(0.104*Cl*Temp*UVA)
# Dominguez-Tello et al. 2017

PP.modeldata_DS <- data.frame(matrix(data = NA, nrow = runs, ncol = 1))
colnames(PP.modeldata_DS) <- c("Model_14")
PP.modeldata_DS[,1] <- 21.4+(36.9*Cl)+(0.986*THM)+(0.59*TOC)+
  (-1.83*Temp)+(-1.21*(TOC-4.1)*(Temp-18.7))
# Salehi et al. 2011
}
}

##### DWTP Model outputs #####
# Graph DWTP model outputs in new window
# Prepare data for ggplot
THM.wtp_fit_df <- data.frame(x = density(THM.wtp_fit)$x, y = density(THM.wtp_fit)$y,
  Model = "THM data")
models_df <- NULL
for (i in 1:8) {
  temp_df <- data.frame(x = density(DWTP.modeldata_TP[[paste0("Model_", i)]])$x,
    y = density(DWTP.modeldata_TP[[paste0("Model_", i)]])$y,
    Model = paste0("Model_", i))
  models_df <- rbind(models_df, temp_df)
}
all_data <- rbind(THM.wtp_fit_df, models_df)

# Create ggplot
p <- ggplot(all_data, aes(x = x, y = y, color = Model, shape = Model)) +
  geom_line(aes(linetype = Model)) +
  scale_color_manual(values = c("firebrick3", "darkorange2", "orange", "dodgerblue3",
    "darkorchid4", "gray49", "darkolivegreen4", "violet", "black")) +
  scale_linetype_manual(values = c("solid", "dotted", "dashed", "dotdash", "longdash",
    "twodash", "solid", "dashed", "solid")) +
  labs(x = "THM [µg/L]", y = "Density") +
  scale_x_continuous(limits = c(0, 150), expand = c(0,0), oob = oob_keep) +
  scale_y_continuous(limits = c(0, 0.06), expand = c(0,0), oob = oob_keep) +
  theme_minimal() +
  theme(legend.title = element_blank(),
    legend.text = element_text(face = "bold"),
    panel.background = element_rect(fill = "white"))

# Print the plot
quartz() # quartz() is a mac specific function, for windows use windows()
print(p)

```

```

# Get DWTP model stats
DWTP.modstats_TP <- get_stats(DWTP.modeldata_TP)
rownames(DWTP.modstats_TP) <- c("1", "2", "3", "4", "5", "6", "7", "8")
DWDS.modstats_TP <- get_stats(DWDS.modeldata_TP)
rownames(DWDS.modstats_TP) <- c("9", "10", "11", "12", "13")

##### DWDS Model outputs #####
# Graph DWDS model outputs in new window
# Prepare data for ggplot
THM.ds_fit_df <- data.frame(x = density(THM.ds_fit)$x, y = density(THM.ds_fit)$y,
                           Model = "THM data")
models_df <- NULL
for (i in 9:13) {
  temp_df <- data.frame(x = density(DWDS.modeldata_DS[[paste0("Model_", i)]])$x,
                       y = density(DWDS.modeldata_DS[[paste0("Model_", i)]])$y,
                       Model = paste0("Model ", i))
  models_df <- rbind(models_df, temp_df)
}
PP_df <- data.frame(x = density(PP.modeldata_DS[[paste0("Model_14")]])$x,
                   y = density(PP.modeldata_DS[[paste0("Model_14")]])$y, Model = "Model 14")
all_data2 <- rbind(THM.ds_fit_df, models_df, PP_df)
all_data2$Model <- factor(all_data2$Model, levels = c("Model 9", "Model 10", "Model 11",
                                                    "Model 12", "Model 13", "Model 14", "THM data"))

# Create ggplot
p2 <- ggplot(all_data2, aes(x = x, y = y, color = Model, shape = Model)) +
  geom_line(aes(linetype = Model)) +
  scale_color_manual(values = c("firebrick1", "darkorange", "darkolivegreen3", "dodgerblue1",
                                "darkorchid4", "gray49", "black")) +
  scale_linetype_manual(values = c("solid", "dotted", "dashed", "dotdash", "longdash",
                                   "twodash", "solid", "dashed", "solid")) +
  labs(x = "THM [ $\mu\text{g/L}$ ]", y = "Density") +
  scale_x_continuous(limits = c(0, 150), expand = c(0,0), oob = oob_keep) +
  scale_y_continuous(limits = c(0, 0.06), expand = c(0,0), oob = oob_keep) +
  theme_minimal() +
  theme(legend.title = element_blank(),
        legend.text = element_text(face = "bold"),
        panel.background = element_rect(fill = "white"))

# Print the plot
quartz() # quartz() is a mac specific function, for windows use windows()
print(p2)

# Get DWDS model stats
DWTP.modstats_DS <- get_stats(DWTP.modeldata_DS)
rownames(DWTP.modstats_DS) <- c("1", "2", "3", "4", "5", "6", "7", "8")
DWDS.modstats_DS <- get_stats(DWDS.modeldata_DS)
rownames(DWDS.modstats_DS) <- c("9", "10", "11", "12", "13")

##### PP Model outputs #####
# Get PP model stats
PP.modstats_DS <- get_stats(PP.modeldata_DS)
rownames(PP.modstats_DS) <- c("14")
DWDS.modstats_DS <- rbind(DWDS.modstats_DS, PP.modstats_DS)

```

```
##### View desc stats #####
# Print statistics
View(DWTP.varstats)
View(DWDS.varstats)
View(DWTP.modstats_TP)
View(DWDS.modstats_TP)
View(DWTP.modstats_DS)
View(DWDS.modstats_DS)

##### View model output comparison #####
DWTP.modelperf <- compare_moments_weighted(DWTP.modeldata_TP, THM.wtp_fit,
                                           THM.ds_fit, weights = c(1,0.1,1))
DWDS.modelperf <- compare_moments_weighted(cbind(DWDS.modeldata_DS, PP.modeldata_DS),
                                           THM.wtp_fit, THM.ds_fit, weights = c(1,0.1,1))

modperf.compare <- cbind(DWTP.modelperf$weighted_difference,
                        DWDS.modelperf$weighted_difference)
modperf.ordered_indices <- order(modperf.compare[, ])
modperf.compare_ordered <- modperf.compare[, modperf.ordered_indices]

View(t(modperf.compare_ordered))
```