

```
# Intro to drc using example pesticide mortality data. Limits of min 0 max 1
# Function section is for example of all possible models. "names" gives the variables in the model the
# proper names for most of them. Good resource, otherwise it's just letters.
# DRC is very good at comparing a range of models and is easy to use.
# The drawback is that you cannot use a model more complex than "response ~ explanatory",
# so no covariates or random variables allowed. If you need to incorporate other variables, look
elsewhere.
```

```
library(drc)
library(tidyverse)
library(xlsx)
```

```
setwd ("Insert your folder here")
```

```
data <- (read.xlsx("drc data.xlsx", 1, header=TRUE))
```

```
ggplot(data, aes(x=Trt, y=Mortality))+
  geom_point()
```

```
## Functions (including names for all coefficients) -----
## Log-logistic models
```

```
ll.4 <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = LL.4(names = c("Slope", "Lower Limit", "Upper Limit", "ED50")),
      type="binomial")
}
```

```
ll.3 <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = LL.3(names = c("Slope", "Upper Limit", "ED50")), ## LL3 fixes lower to zero
      type="binomial")
}
```

```
ll.3u <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = LL.3u(names = c("Slope", "Lower Limit", "ED50")),
      type="binomial")
}
```

```
ll.2 <- function(df){
  drm(Mortality ~ Trt, weights = Count,
      data = df,
```

```

    fct = LL.2(names = c("Slope", "ED50")),
    type="binomial")
}

```

#Asymptotic regression models

```

AR.2 <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = AR.2(names = c("Upper limit", "Slope")),
      type="binomial")
}

```

```

AR.3 <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = AR.3(names = c("Lower limit", "Upper limit", "Slope")),
      type="binomial")
}

```

#Weibull models

```

w1.2 <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = W1.2(),
      type="binomial")
}

```

```

w1.2 <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = W1.2(names = c("Something", "Inflection point")),
      type="binomial")
}

```

```

w2.2 <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = W2.2(names = c("????", "Inflection point")),
      type="binomial")
}

```

```

w1.3 <- function(df){
  drm(Mortality ~ Trt, weights=Count,

```

```

    data = df,
    fct = W1.3(names = c("????", "Upper limit", "Inflection point")),
    type="binomial")
}

w2.3 <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = W2.3(names = c("????", "Upper limit", "Inflection point")),
      type="binomial")
}

w1.3u <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = W1.3u(names = c("????", "Lower limit", "Inflection point")),
      type="binomial")
}

w2.3u <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = W2.3u(names = c("Slope of infelction", "Lower limit", " Log of inflection point")),
      type="binomial")
}

w1.4 <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = W1.4(names = c("????", "Lower limit", "Upper limit", "Inflection point")),
      type="binomial")
}

w2.4 <- function(df){
  drm(Mortality ~ Trt, weights=Count,
      data = df,
      fct = W2.4(names = c("????", "Lower limit", "Upper limit", "Inflection point")),
      type="binomial")
}

getMeanFunctions()

```

Breaking down the basics -----

#basic layout:

```

# drm(response_variable ~ explanatory_variable, weights = total_samples_in_rep,
#      data=data_object,

```

```
#      fct = model ( names = specific for each model)),  
#      type = "distribution_of_data")
```

```
#Log-logistic model with four parameters, lower limit set to 0 upper limit set to 1  
model.LL.2 <- drm(Mortality ~ Trt, weights = Count,  
  data = data,  
  fct = LL.2(names = c("Slope", "ED50")),  
  type="binomial")
```

```
model.AR.2 <- drm(Mortality ~ Trt, weights=Count,  
  data = data,  
  fct = AR.2(names = c("Upper limit", "Slope")),  
  type="binomial")
```

```
#Weibull model with three parameters, upper limit set to 1  
model.W2.3u <- drm(Mortality ~ Trt, weights=Count,  
  data = data,  
  fct = W2.3u(names = c("Slope of infelction", "Lower limit", " Log of inflection point")),  
  type="binomial")
```

```
#quick and dirty plot  
plot(model.LL.2)  
plot(model.AR.2)  
plot(model.W2.3u)
```

```
# assesses model fit using pearson's Chi squared  
modelFit(model.LL.2)  
modelFit(model.AR.2)  
modelFit(model.W2.3u)
```

```
# compare AIC  
AIC(model.LL.2)  
AIC(model.AR.2)  
AIC(model.W2.3u)
```

```
# comparing residuals and fitted values  
op <- par(mfrow = c(1, 2)) #put two graphs together  
plot(residuals(model.LL.2) ~ fitted(model.LL.2), main="Residuals vs Fitted")
```

```
abline(h=0)
qqnorm(residuals(model.LL.2))
qqline(residuals(model.LL.2))
```

```
op <- par(mfrow = c(1, 2)) #put two graphs together
plot(residuals(model.AR.2) ~ fitted(model.AR.2), main="Residuals vs Fitted")
abline(h=0)
qqnorm(residuals(model.AR.2))
qqline(residuals(model.AR.2))
```

```
op <- par(mfrow = c(1, 2)) #put two graphs together
plot(residuals(model.W2.3u) ~ fitted(model.W2.3u), main="Residuals vs Fitted")
abline(h=0)
qqnorm(residuals(model.W2.3u))
qqline(residuals(model.W2.3u))
```

```
# comparing effective doses (median lethal dose = 50)
ED.LL.2 <- ED(model.LL.2, c(50), interval = "delta")
ED.AR.2 <- ED(model.AR.2, c(50), interval = "delta")
ED.W2.3u <- ED(model.W2.3u, c(50), interval = "delta")
```

```
# ^ Function EDcomp useful for comparing values that are less obvious than this
```

```
## Plotting the models much more nicely -----
```

```
newdata <- expand.grid(Trt=seq(1, 150, length=200)) #generates predicted curve in combination with
predict()
```

```
pred<-predict(model.LL.2, newdata=newdata, interval="confidence") %>% data.frame()
pred$x <- newdata
```

```
#first 10 lines are for aesthetics
```

```
ggplot()+
  xlab ("Trt") + ylab ("Mortality") + ggtitle ("LL.2")+
  theme_bw()+
  theme(axis.line = element_line(colour = "black"),
        axis.line.x = element_line(colour = "black"),
        axis.line.y = element_line(colour = "black"),
        panel.border = element_blank(),
        panel.background = element_blank(),
        plot.title=element_text(size=25),
        axis.title=element_text(size=25),
        axis.text=element_text(size=16))+
```

```

# predicted curve and confidence bands
geom_line(data=pred, aes(x=(x),y=Prediction))+
geom_ribbon(data=pred, aes(x=(x), y=Prediction,ymin=Lower, ymax=Upper), alpha=0.2)+
# plotting original data
geom_point(data=data, aes(x=(Trt), y=Mortality))+
# bar with 95% confidence intervals for LD50
geom_vline(xintercept=ED.LL.2[1,1], colour = "black",size=1)+
geom_rect(aes(xmin=ED.LL.2[1,3], xmax=ED.LL.2[1,4],
              ymin=-Inf, ymax=Inf),fill="black", alpha=0.2)

```

```

newdata <- expand.grid(Trt=seq(1, 150, length=200)) #generates predicted curve in combination with
predict()

```

```

pred<-predict(model.AR.2, newdata=newdata, interval="confidence") %>% data.frame()
pred$x <- newdata

```

```

#first 10 lines are for aesthetics
ggplot()+
  xlab ("Trt") + ylab ("Mortality") + ggtitle ("AR.3")+
  theme_bw()+
  theme(axis.line = element_line(colour = "black"),
        axis.line.x = element_line(colour = "black"),
        axis.line.y = element_line(colour = "black"),
        panel.border = element_blank(),
        panel.background = element_blank(),
        plot.title=element_text(size=25),
        axis.title=element_text(size=25),
        axis.text=element_text(size=16))+
# predicted curve and confidence bands
geom_line(data=pred, aes(x=(x),y=Prediction))+
geom_ribbon(data=pred, aes(x=(x), y=Prediction,ymin=Lower, ymax=Upper), alpha=0.2)+
# plotting original data
geom_point(data=data, aes(x=(Trt), y=Mortality))+
# bar with 95% confidence intervals for LD50
geom_vline(xintercept=ED.AR.2[1,1], colour = "black",size=1)+
geom_rect(aes(xmin=ED.AR.2[1,3], xmax=ED.AR.2[1,4],
              ymin=-Inf, ymax=Inf),fill="black", alpha=0.2)

```

```
newdata <- expand.grid(Trt=seq(1, 150, length=200)) #generates predicted curve in combination with
predict()
```

```
pred<-predict(model.W2.3u, newdata=newdata, interval="confidence") %>% data.frame()
pred$x <- newdata
```

```
#first 10 lines are for aesthetics
```

```
ggplot()+
  xlab ("Trt") + ylab ("Mortality") + ggtitle ("W2.3u")+
  theme_bw()+
  theme(axis.line = element_line(colour = "black"),
        axis.line.x = element_line(colour = "black"),
        axis.line.y = element_line(colour = "black"),
        panel.border = element_blank(),
        panel.background = element_blank(),
        plot.title=element_text(size=25),
        axis.title=element_text(size=25),
        axis.text=element_text(size=16))+
```

```
# predicted curve and confidence bands
```

```
geom_line(data=pred, aes(x=(x),y=Prediction))+
geom_ribbon(data=pred, aes(x=(x), y=Prediction,ymin=Lower, ymax=Upper), alpha=0.2)+
```

```
# plotting original data
```

```
geom_point(data=data, aes(x=(Trt), y=Mortality))+
```

```
# bar with 95% confidence intervals for LD50
```

```
geom_vline(xintercept=ED.W2.3u[1,1], colour = "black",size=1)+
geom_rect(aes(xmin=ED.W2.3u[1,3], xmax=ED.W2.3u[1,4],
              ymin=-Inf, ymax=Inf),fill="black", alpha=0.2)
```