d-r-sim

Chris Hoover

March 3, 2018

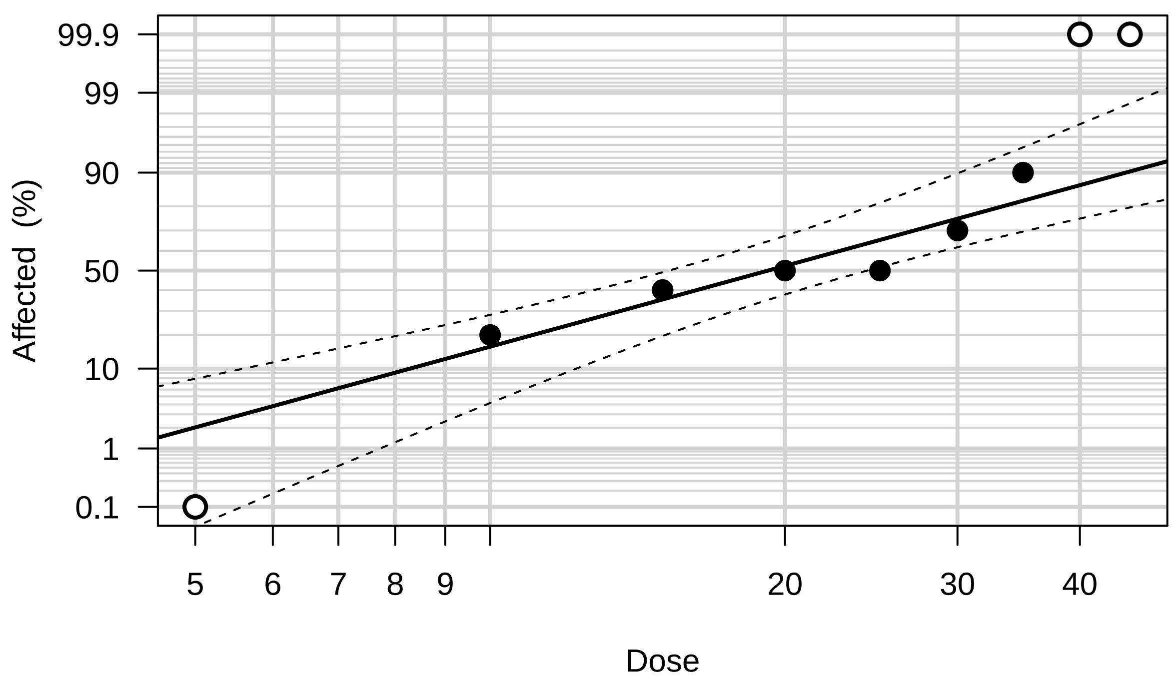
Trying to determine how best to reproduce a dose-response function when provided with LC parameters and a slope function generated using the method of Litchfield and Wilcoxon which fits a linear response to log10(dose) – probit(response) data.

First generate data and fit d-r response using the `LW1949` package

```
dr <-
  dataprep(dose = seq(0, 45, 5),
           nfx = c(0, 0, 2, 4, 5, 5, 7, 9, 10, 10),
           ntot = rep(10,10))

intslope <- fitLWauto(dr)

fLW <- LWestimate(intslope, dr)
```

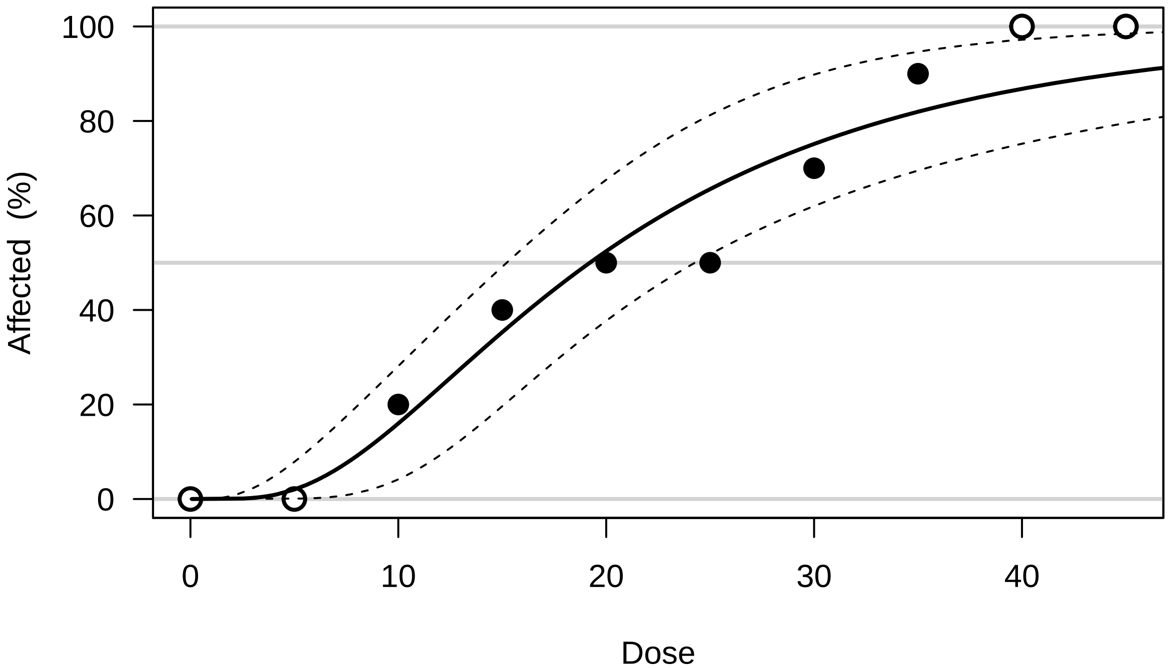
Plot the linear model fit to transformed data

```
plotDELP(dr)
  predLinesLP(fLW)
```



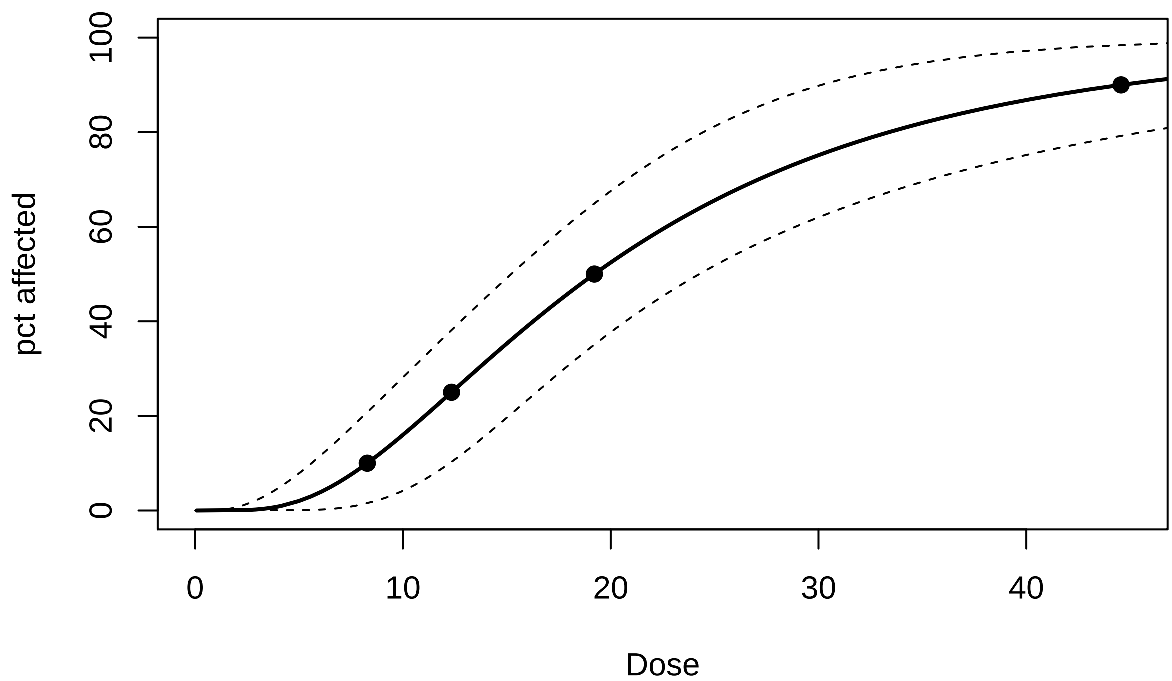
Plot the fit to the untransformed data

```
plotDE(dr)
  predLines(fLW)
```



Get estimates of LC_{10}, LC_{25}, LC_{50}, LC_{90} and plot. Ideally will be able to reproduce the d-r function from only the LC_{50} and slp parameters

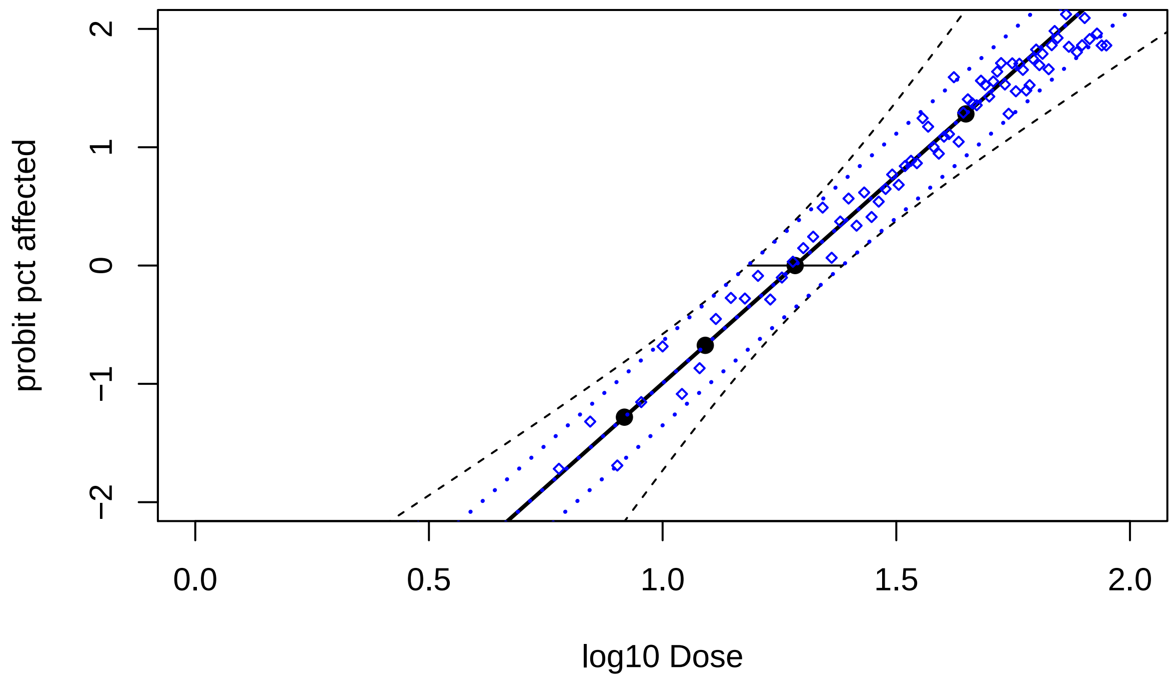
```
pars <- as.data.frame(predlinear(c(10, 25, 50, 90), fLW))
  pars$log10_ED <- log10(pars$ED)
  pars$probit <- qnorm(pars$pct/100)

lc50 <- predlinear(50, fLW)
slp <- fLW$LWest["S"]

plot(pars[,2], pars[,1], pch = 16, cex = 1.2, xlim = c(0,45), ylim = c(0,100),
     xlab = "Dose", ylab = "pct affected")
  predLines(fLW)
```



Can we reproduce the D-R function with uncertainty from LC_{10}, LC_{25}, LC_{50}, LC_{90} and CI of LC_{50}?

```
plot(log10(pars[,2]), qnorm(pars[,1]/100), pch = 16, cex = 1.2, xlim = c(0,2), ylim = c(-2,2),
     xlab = "log10 Dose", ylab = "probit pct affected")
  segments(y0 = qnorm(0.5), y1 = qnorm(0.5),
           x0 = log10(pars$lower[pars$pct == 50]), x1 = log10(pars$upper[pars$pct == 50]))
  predLinesLP(fLW)

#Fit linear model to the LC values provided
lc_mod <- lm(probit ~ log10_ED, data = pars)

#Plot results including uncertainty of LC
d_fx <- function(d, s, lc50){
  pnorm(s * log10(d/lc50))
}

lines(log10(c(0:100)), qnorm(d_fx(d = c(0:100), s = coef(lc_mod)[2], lc50 = lc50[2])),
      lty = 3, col = 4, lwd = 2)
lines(log10(c(0:100)), qnorm(d_fx(d = c(0:100), s = coef(lc_mod)[2], lc50 = lc50[3])),
      lty = 3, col = 4, lwd = 2)
lines(log10(c(0:100)), qnorm(d_fx(d = c(0:100), s = coef(lc_mod)[2], lc50 = lc50[4])),
      lty = 3, col = 4, lwd = 2)

#What if we sample from lc50 range?
lc50_se <- log10(lc50[4]/lc50[2]) / 1.96
```

```r
d_fx_uncertainty <- function(d, s, lc50, se){
  lc_use = 10^(rnorm(1, log10(lc50), se))
  pnorm(s * log10(d/lc_use))
}

set.seed(43093)
points(log10(c(0:100)), qnorm(sapply(c(0:100), d_fx_uncertainty, s = coef(lc_mod)[2], lc50 = lc50[2], se
       pch = 5, col = 4, cex = 0.5)
```

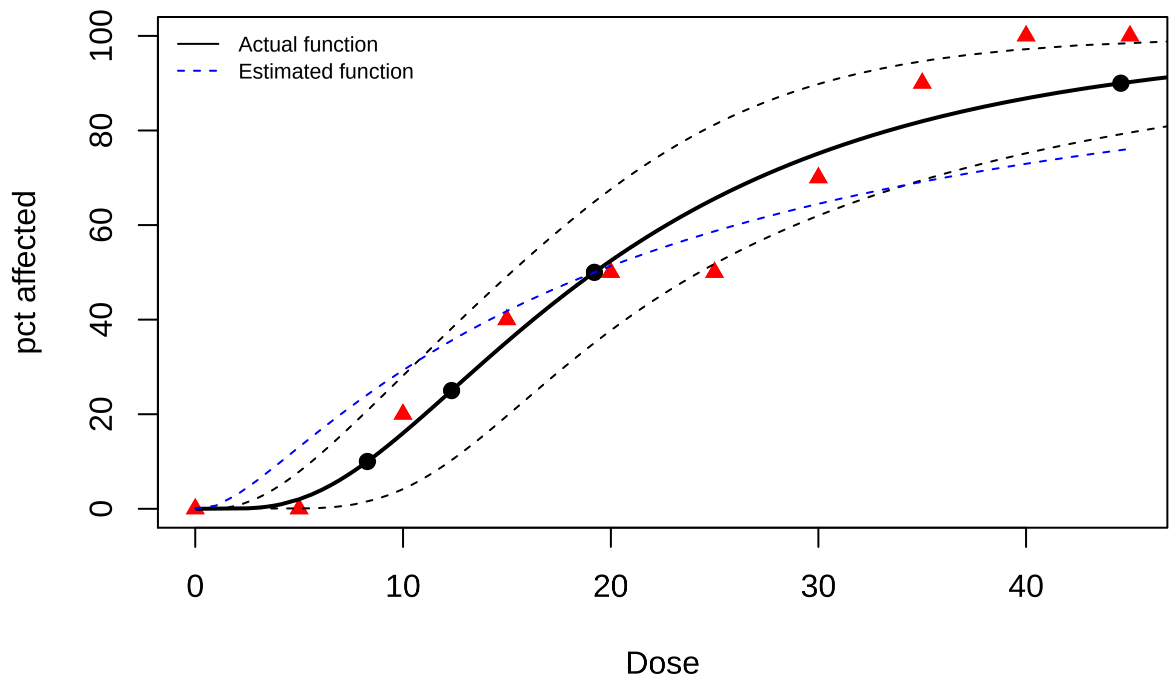


Doesn't do a perfect job of reproducing uncertainty, but let's see what else we can do

Let's see what we can do given just the LC_{50} and *slp*

```r
lc50 <- predlinear(50, fLW)

slp <- fLW$LWest["S"]

lc84 <- predlinear(84, fLW)
lc16 <- predlinear(16, fLW)

slp_man <- (lc84[2]/lc50[2] + lc50[2]/lc16[2])/2
#Estimate intercept of the linear model using slope as true linear slope and lc50 as a reference point
intercept = qnorm(.5) - slp*log10(lc50[2])

#d-r function using the estimated intercept and slope
dr_fx <- function(d, s, lc50){
```

```
  r = pnorm(s * log10(d/lc50))
  r
}

plot(pars[,2], pars[,1], pch = 16, cex = 1.2, xlim = c(0,45), ylim = c(0,100),
     xlab = "Dose", ylab = "pct affected")
  points(dr$dose, (dr$nfx/dr$ntot)*100, pch = 17, col = 2)
  predLines(fLW)

  lines(c(0:45), dr_fx(c(0:45), s = slp, lc50 = lc50[2])*100, lty = 2, col = 4)
  legend("topleft", legend = c("Actual function", "Estimated function"), lty = c(1,2), col = c(1,4), bty
```



We can estimate the slope of the underlying linear model, b_1, given the slope parameter, S using some algebra:

$$S = \frac{\frac{LC_{84}}{LC_{50}} + \frac{LC_{50}}{LC_{16}}}{2} \quad (1)$$

$$0 = 2 * S * LC_{50} LC_{16} - LC_{84} LC_{16} - LC_{50}^2 \quad (2)$$

We also know:

$$b_1 = \frac{probit(0.84) - probit(0.5)}{log_{10}(LC_{84}) - log_{10}(LC_{50})} \quad (3)$$

$$b_1 = \frac{probit(0.5) - probit(0.16)}{log_{10}(LC_{0.5}) - log_{10}(LC_{0.16})} \quad (4)$$

Solving for LC_{84}:

$$LC_{84} = LC_{50}10^{\frac{probit(0.84)}{b_1}} \tag{5}$$

Solving for LC_{16}:

$$LC_{16} = \frac{LC_{50}}{10^{\frac{-probit(0.16)}{b_1}}} \tag{6}$$

Substituting for LC_{84} and LC_{16} we have:

$$0 = 2 * S * LC_{50}\frac{LC_{50}}{10^{\frac{-probit(0.16)}{b_1}}} - LC_{50}10^{\frac{probit(0.84)}{b_1}}\frac{LC_{50}}{10^{\frac{-probit(0.16)}{b_1}}} - LC_{50}^2 \tag{7}$$

Dividing both sides by LC_{50}^2 we get:

$$0 = \frac{2S}{10^{\frac{-probit(0.16)}{b_1}}} - \frac{10^{\frac{probit(0.84)}{b_1}}}{10^{\frac{-probit(0.16)}{b_1}}} - 1 \tag{8}$$

We can then estimate b_1 given S by solving the above equation

$$\tag{9}$$

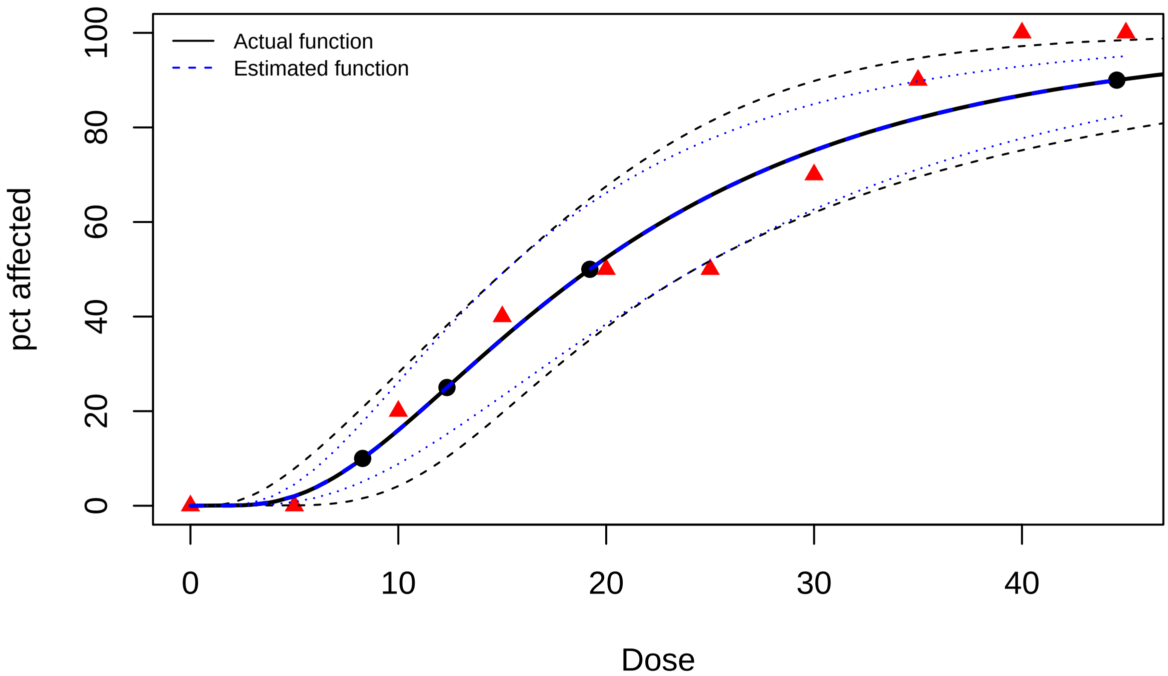
```
get_b1 <- function(slp){
  uniroot.all(f = function(b1){2*slp / 10^(-qnorm(.16)/b1) - (10^(qnorm(.84)/b1) / 10^(-qnorm(.16)/b1))
}

get_b1(slp = slp)
```

```
## [1] 3.507909
```

```
intslope[2]
```

```
##    Slope
## 3.507905
```

```
b1<- get_b1(slp = slp)

dr_fx_b1 <- function(d, b1, lc50){
    pnorm(b1 * log10(d/lc50))
  }


plot(pars[,2], pars[,1], pch = 16, cex = 1.2, xlim = c(0,45), ylim = c(0,100),
     xlab = "Dose", ylab = "pct affected")
  points(dr$dose, (dr$nfx/dr$ntot)*100, pch = 17, col = 2)
  predLines(fLW)

  lines(c(0:45), dr_fx_b1(c(0:45), b1 = b1, lc50 = lc50[2])*100, lty = 2, col = 4, lwd = 2)
  lines(c(0:45), dr_fx_b1(c(0:45), b1 = b1, lc50 = lc50[3])*100, lty = 3, col = 4)
  lines(c(0:45), dr_fx_b1(c(0:45), b1 = b1, lc50 = lc50[4])*100, lty = 3, col = 4)
legend("topleft", legend = c("Actual function", "Estimated function"), lty = c(1,2), col = c(1,4), bty
```



Doesn't do a perfect job of reproducing uncertainty because only variability comes from 95% CI of LC_{50} parameter