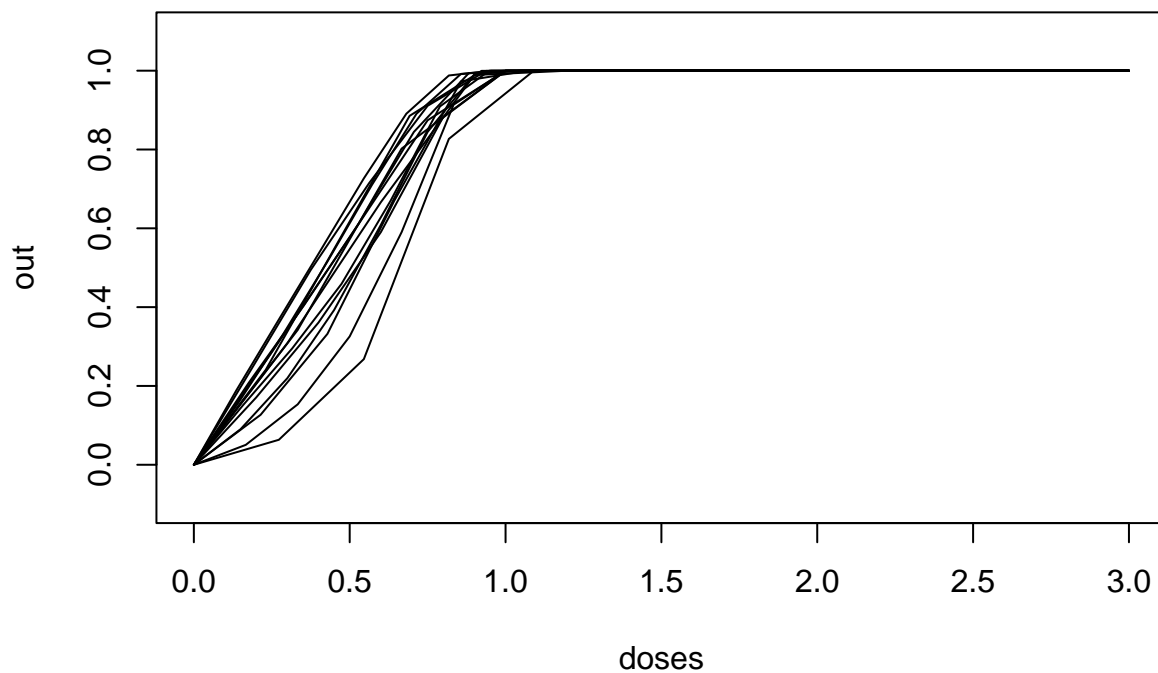


Dichotomous Models Test Codes

2023-09-07

Multistage

```
# maximum number of dose: 23
for (n in 10:23){
  k <- n
  doses <- seq(0,3,length.out = k)
  b <- runif(n = k,min = 0,max = 1.5) %>% round(., 3) # dose coefficients no upper limit
  out <- multistage(b,x = doses)
  if(n==10){
    plot(doses,out,type = "l",ylim = c(-0.1,1.1))
  } else{
    lines(doses,out)
  }
}
}
```



weibull

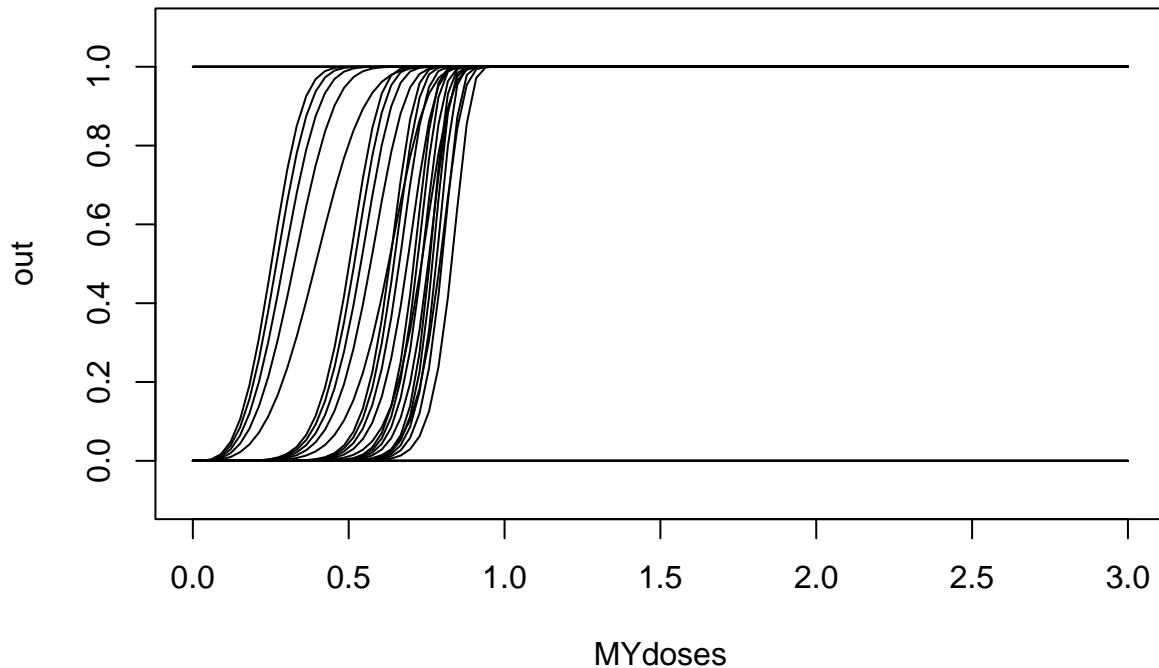
```
# Randomly select 3 parameters within the allowable parameter bounds
a <- runif(n = 3,min = 0,max = 18) %>% round(.,3)
b <- runif(n = 3,min = 0,max = 100) %>% round(.,3)
wparams <- expand.grid(a,b)

for(i in 1:nrow(wparams)){
  out <- weibull(ps = unlist(wparams[i,]),x = MYdoses)
  plot(MYdoses,out,sub = paste(wparams[i,],collapse = ", "),type = "l")
}

# Use parameters across the range of allowable values for each parameter
a <- seq(0,18,length.out = 6)
b <- seq(0,99,length.out = 6)

wparams <- expand.grid(a,b)

for(i in 1:nrow(wparams)){
  out <- weibull(ps = unlist(wparams[i,]),x = MYdoses)
  if(i==1){
    plot(MYdoses,out,type = "l",ylim = c(-0.1,1.1))
  }else{
    lines(MYdoses,out)
  }
}
```



Gamma

Derivation: Formula in BMDS User Guide: $g + \frac{(1-g)}{\Gamma(a)} \left[\int_0^{bx} t^{(a-1)} e^{-t} dt \right]$

Inside the bracket is just the CDF of the regularized gamma function: $g + (1-g) \left[\frac{1}{\Gamma(a)} \int_0^{bx} t^{(a-1)} e^{-t} dt \right]$

For now we assume background $g = 0$, so Dichotomous Gamma is just using `pgamma`. From there we have:
`$g + (1-g)* pgamma(x, a, b) $`

```
# Randomly select 3 parameters within the allowable parameter bounds
a <- runif(n = 3,min = 0.2,max = 18) %>% round(.,3)
b <- runif(n = 3,min = 0,max = 100) %>% round(.,3)
wparams <- expand.grid(a,b)

for(i in 1:nrow(wparams)){
  out <- gamma_d(ps = unlist(wparams[i,]),x = MYdoses)
  plot(MYdoses,out,sub = paste(wparams[i,],collapse = ", "),type = "l")
}
```

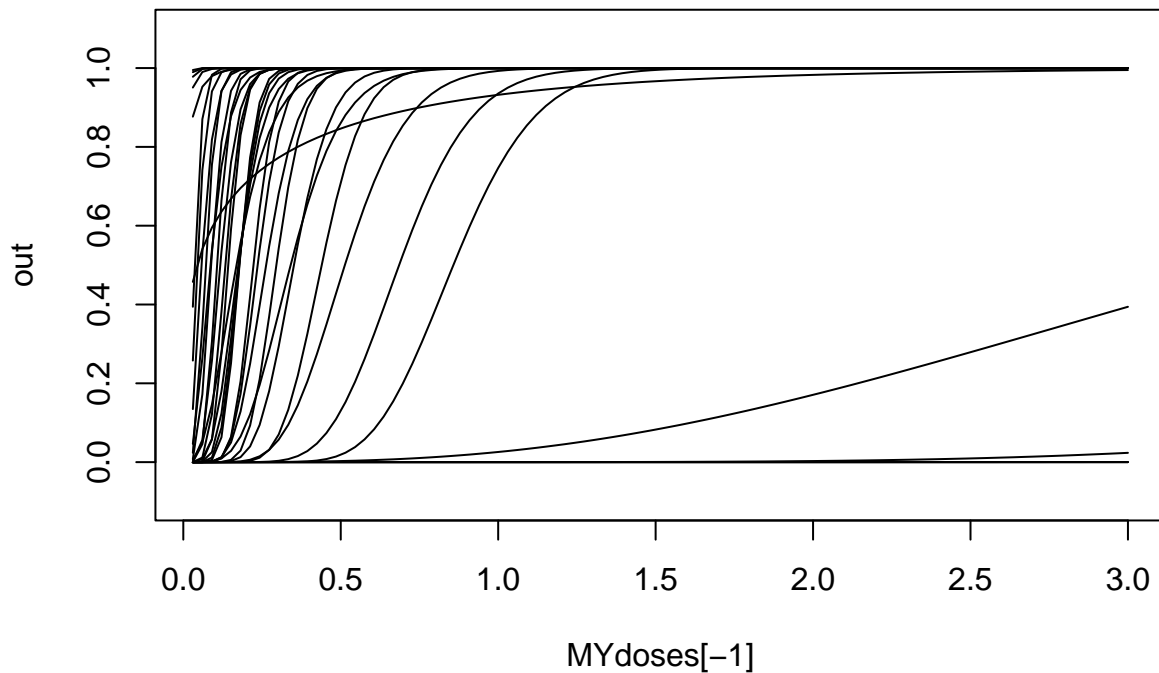
```
# Use parameters across the range of allowable values for each parameter
a <- seq(0.25,18,length.out = 6)
b <- seq(1,99,length.out = 6)

wparams <- expand.grid(a,b)
```

```

for(i in 1:nrow(wparams)){
  out <- gamma_d(ps = unlist(wparams[i,]),x = MYdoses[-1])
  if(i==1){
    plot(MYdoses[-1],out,type = "l",ylim = c(-0.1,1.1))
  }else{
    lines(MYdoses[-1],out)
  }
}
}

```



logistic

```

# Randomly select 3 parameters within the allowable parameter bounds
a <- runif(n = 3,min = -18,max = 18) %>% round(.,3)
b <- runif(n = 3,min = 0,max = 100) %>% round(.,3)
wparams <- expand.grid(a,b)

for(i in 1:nrow(wparams)){
  out <- logistic(ps = unlist(wparams[i,]),x = MYdoses)
  plot(MYdoses,out,sub = paste(wparams[i,],collapse = ", "),type = "l")
}

```

```

# Use parameters across the range of allowable values for each parameter
a <- seq(-17.5,17.5,length.out = 6)

```

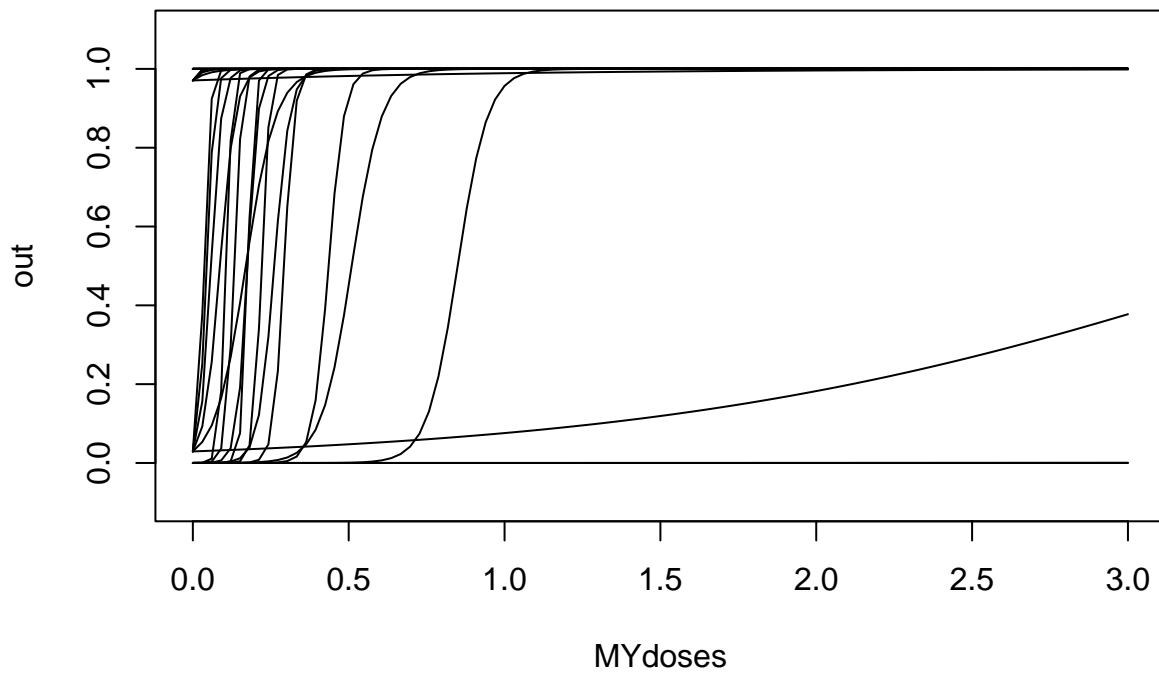
```

b <- seq(1,99,length.out = 6)

wparams <- expand.grid(a,b)

for(i in 1:nrow(wparams)){
  out <- logistic(ps = unlist(wparams[i,]),x = MYdoses)
  if(i==1){
    plot(MYdoses,out,type = "l",ylim = c(-0.1,1.1))
  }else{
    lines(MYdoses,out)
  }
}

```



log-logistic

```

# Randomly select 3 parameters within the allowable parameter bounds
a <- runif(n = 3,min = -18,max = 18) %>% round(.,3)
b <- runif(n = 3,min = 0,max = 18) %>% round(.,3)
wparams <- expand.grid(a,b)

for(i in 1:nrow(wparams)){
  out <- llogistic(ps = unlist(wparams[i,]),x = MYdoses)
  plot(MYdoses,out,sub = paste(wparams[i,],collapse = ", "),type = "l")
}

```

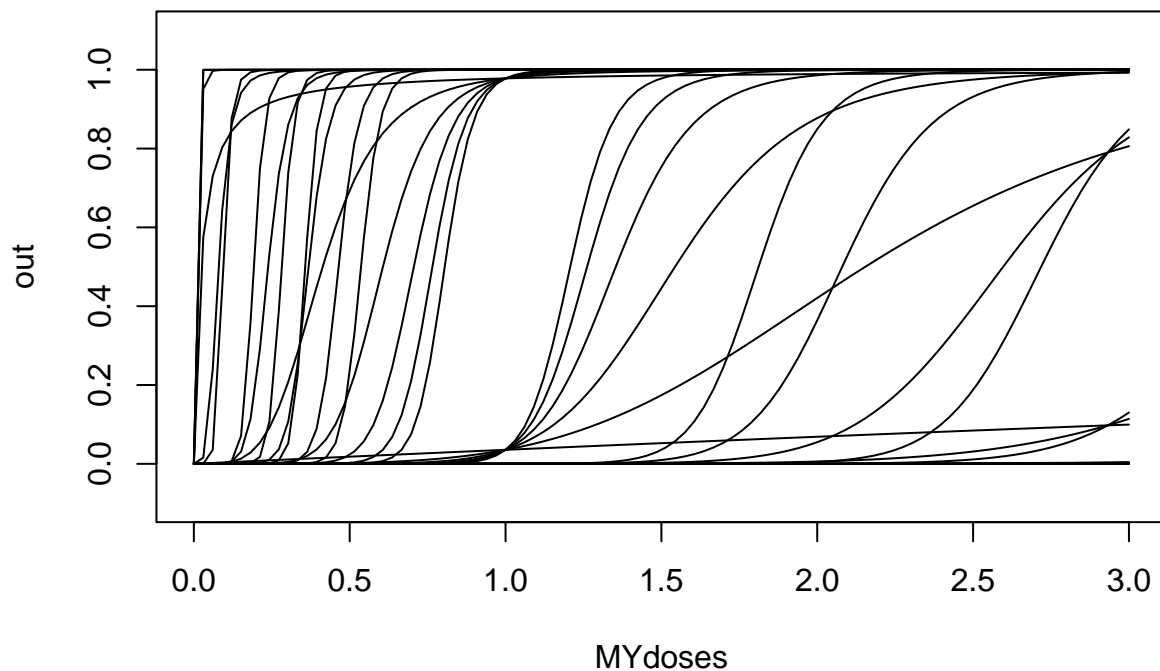
```

# Use parameters across the range of allowable values for each parameter
a <- seq(-17.5,18,length.out = 6)
b <- seq(1,17.5,length.out = 6)

wparams <- expand.grid(a,b)

for(i in 1:nrow(wparams)){
  out <- llogistic(ps = unlist(wparams[i,]),x = MYdoses)
  if(i==1){
    plot(MYdoses,out,type = "l",ylim = c(-0.1,1.1))
  }else{
    lines(MYdoses,out)
  }
}

```



Probit

Derivation: Formula in BMDS User Guide: $\int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$ where $x = a + b * dose$

The integral is just standard Normal cumulative distribution function, so we can `pnorm` and input the updated x. Same with log-probit except it takes natural log of x.

```
pnorm(a + b * dose)
```

```

# Randomly select 3 parameters within the allowable parameter bounds
a <- runif(n = 3,min = -18,max = 18) %>% round(.,3)
b <- runif(n = 3,min = 0,max = 18) %>% round(.,3)
wparams <- expand.grid(a,b)

for(i in 1:nrow(wparams)){
  out <- probit(ps = unlist(wparams[i,]),x = MYdoses)
  plot(MYdoses,out,sub = paste(wparams[i,],collapse = ", "),type = "l")
}

```

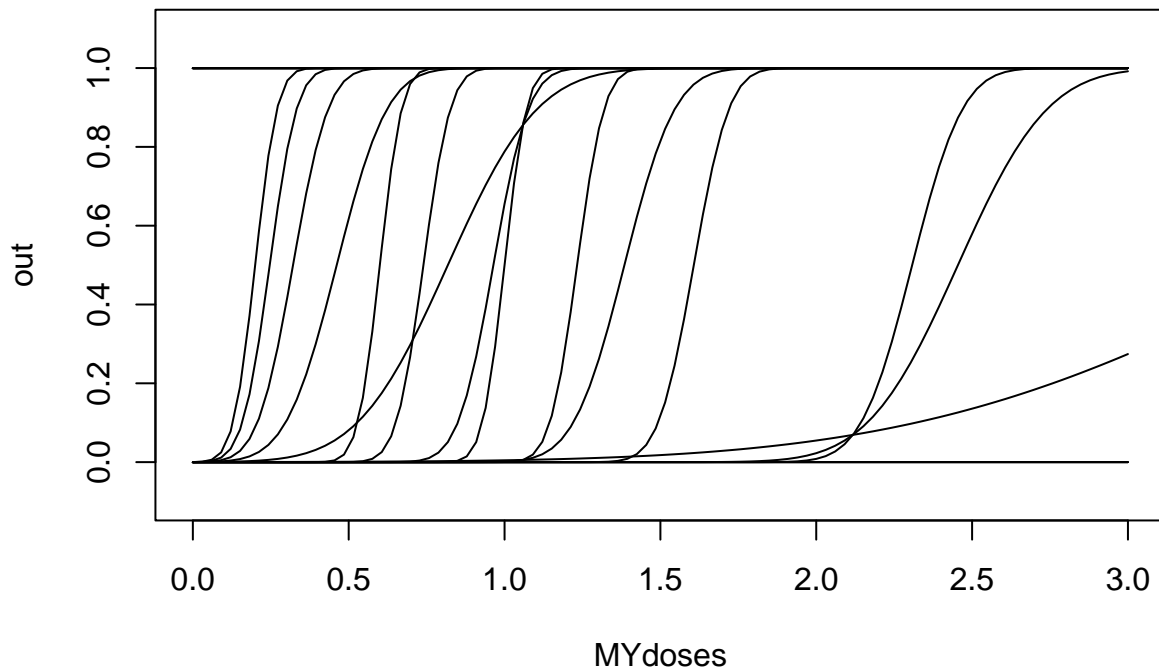
```

# Use parameters across the range of allowable values for each parameter
a <- seq(-18,18,length.out = 6)
b <- seq(1,18,length.out = 6)

wparams <- expand.grid(a,b)

for(i in 1:nrow(wparams)){
  out <- probit(ps = unlist(wparams[i,]),x = MYdoses)
  if(i==1){
    plot(MYdoses,out,type = "l",ylim = c(-0.1,1.1))
  }else{
    lines(MYdoses,out)
  }
}

```



log-probit

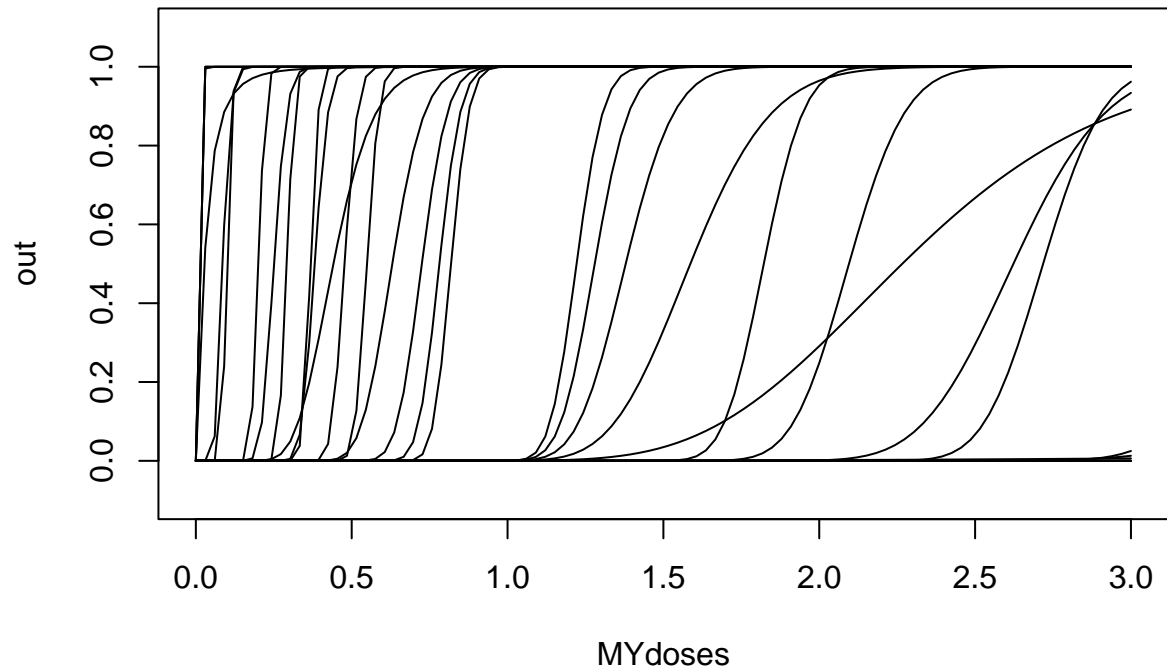
```
# Randomly select 3 parameters within the allowable parameter bounds
a <- runif(n = 3,min = -18,max = 18) %>% round(.,3)
b <- runif(n = 3,min = 0,max = 18) %>% round(.,3)
wparams <- expand.grid(a,b)

for(i in 1:nrow(wparams)){
  out <- lprobit(ps = unlist(wparams[i,]),x = MYdoses)
  plot(MYdoses,out,sub = paste(wparams[i,],collapse = ", "),type = "l")
}

# Use parameters across the range of allowable values for each parameter
a <- seq(-18,18,length.out = 6)
b <- seq(1,18,length.out = 6)

wparams <- expand.grid(a,b)

for(i in 1:nrow(wparams)){
  out <- lprobit(ps = unlist(wparams[i,]),x = MYdoses)
  if(i==1){
    plot(MYdoses,out,type = "l",ylim = c(-0.1,1.1))
  }else{
    lines(MYdoses,out)
  }
}
```

Dichotomous Hill

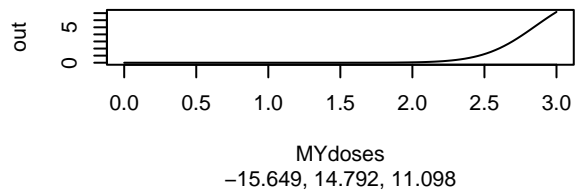
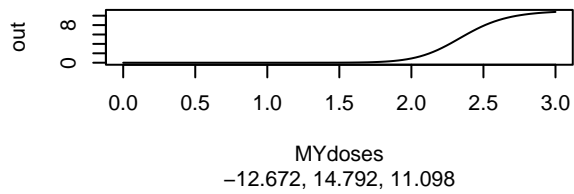
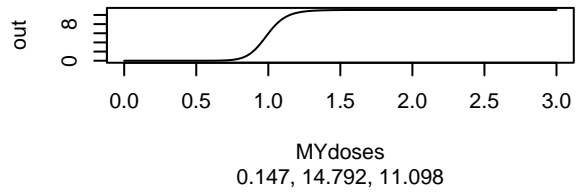
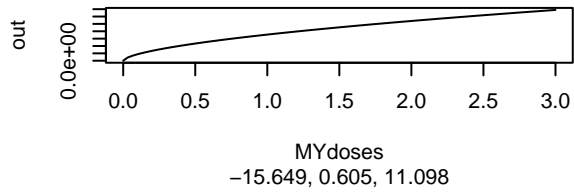
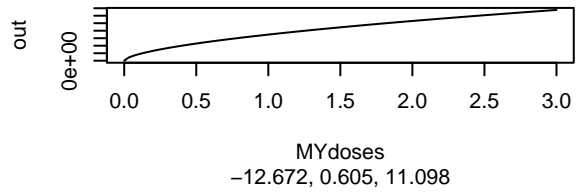
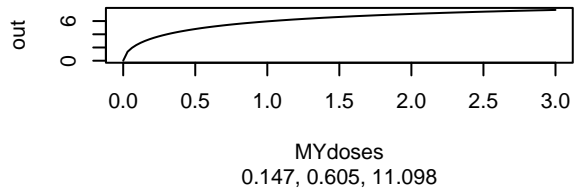
```

# Randomly select 3 parameters within the allowable parameter bounds
a <- runif(n = 3,min = -18,max = 18) %>% round(.,3)
b <- runif(n = 3,min = 0,max = 18) %>% round(.,3)
v <- runif(n = 3,min = -18,max = 18) %>% round(.,3)
wparams <- expand.grid(a,b,v)

par( mfrow= c(3,2) )

for(i in 1:6){
  out <- d_hill(ps = unlist(wparams[i,]),x = MYdoses)
  plot(MYdoses,out,sub = paste(wparams[i,],collapse = ", "),type = "l")
}

```



```
# Use parameters across the range of allowable values for each parameter
a <- seq(-17.5,18,length.out = 6)
b <- seq(1,18,length.out = 6)
v <- seq(-17.5,18,length.out = 6)

wparams <- expand.grid(a,b,v)

for(i in 1:nrow(wparams)){
  out <- d_hill(ps = unlist(wparams[i,]),x = MYdoses)
  if(i==1){
    plot(MYdoses,out,type = "l")
  }else{
    lines(MYdoses,out)
  }
}
```