```
source("ams_initialize_script.R")
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##     smiths

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:GGally':
##
##     nasa

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(RxODE)
library(dplyr)
```

Model F is defined below

```
ivsc_4cmtct_shedct = function() {
  model           = list()
  model$name      = as.character(sys.calls()[[sys.nframe()]])

  #COMPARTMENTS AND INITIAL CONDITIONS
  model$cmtshort  = c('AmtD0','D1','D2','D3','S1','S3','M3','DS1','DS3','DM3','DM1','M1')
  model$init      = function(p){
             init = c(AmtD0=0,D1=0,D2=0,D3=0,S1=0,S3=0,M3=0,DS1=0,DS3=0,DM3=0,DM1=0,M1=0)
             p    = p %>% t() %>% as.data.frame()


             ksyn = with(p,c(ksynS1,ksynS3,ksynM1,ksynM3))
             K    = with(p,matrix(c(keS1+k13S         , -k13S*VS1/VS3       ,  -kshedM1,         0,
                                    -k31S*VS3/VS1 ,  keS3 + k31S        ,  0       ,  -kshedM3,
                                    0            ,  0                  , kshedM1+keM1+k13M, -k31M*VD3/V
                                    0            ,  0                  , -k13M*VD3/VD1,   kshedM3+keM3+
                                  nrow = 4, byrow=TRUE))
             x    = solve(K,ksyn)

             init["S1"] = unlist(x[1])
             init["S3"] = unlist(x[2])
             init["M1"] = unlist(x[3])
             init["M3"] = unlist(x[4])
             return(init)
  }

  #PARAMEETRS IN MODEL
  model$pin       = c('F','ka','VD1','VD2','VD3','VS1','VS3','VDS1','VDS3',
                      'k12D','k21D','k13D','k31D','k13S','k31S','k13DS','k31DS',
                      'ksynS1','ksynS3','ksynM3','keD1','keD3','keS1','keS3','keDS1','keDS3','keM3','ke
                      'kon1','koff1','kon3','koff3',
                      'kshedM3','kshedDM3','ksynM1','kshedM1','kshedDM1','keM1','keDM1','k13M','k31M','
  model$pode      = model$pin

                   #INPUT/SYNTHESIS/SHED   DISTRIBUTION (CENTRAL/TUMOR)      BINDING
  model$rxode.str = '
     D1           = AmtD1/VD1;
     d/dt(AmtD0)  =  -ka *AmtD0;
     d/dt(AmtD1)  =(F*ka *AmtD0/VD1 - k13D *D1 +  k31D *VD3/VD1*D3  - keD1 *D1  - kon1*D1*S1 + koff1*DS
     d/dt(D2)     = k12D*VD1/VD2*D1 - k21D*D2;
     d/dt(D3)     = k13D *VD1/VD3*D1 - k31D*D3  - keD3 *D3  - kon3*D3*(S3+M3) + koff3*(DS3+DM3);
     d/dt(S1)     = ksynS1+kshedM1*M1 - k13S *S1 +  k31S*VS3/VS1*S3  - keS1 *S1  - kon1*D1*S1      + ko
     d/dt(S3)     = ksynS3 +kshedM3*M3   + k13S *VS1/VS3*S1 - k31S*S3  - keS3 *S3  - kon3*D3*S3  + koff
     d/dt(M3)     = ksynM3 -kshedM3*M3  -k31M*M3+k13M*VD1/VD3*M1  - keM3 *M3  - kon3*D3*M3 + koff3*DM3;
     d/dt(DS1)    = kshedM1*DM1 - k13DS*DS1 + k31DS*VDS3/VDS1*DS3 - keDS1*DS1 + kon1*D1*S1 - koff1*DS1;
     d/dt(DS3)    = kshedDM3*DM3 + k13DS*VDS1/VDS3*DS1 - k31DS*DS3 - keDS3*DS3 + kon3*D3*S3  - koff3*DS
     d/dt(DM3)    = -kshedDM3*DM3  - keDM3*DM3 + kon3*D3*M3 - koff3*DM3-k31DM*DM3+k13DM*(VD1/VD3)*DM1;
     d/dt(DM1)    = -keDM1*DM1 -kshedDM1*DM1 +kon1*D1*M1 -koff1*DM1-k13DM*DM1+k31DM*(VD3/VD1)*DM3;
     d/dt(M1)     = ksynM1 -kshedM1*M1 -keM1*M1 +k31M*VD3/VD1*M3 -k13M*M1 -kon1*D1*M1 +koff1*DM1;
  '
  model$rxode     = RxODE(model = model$rxode.str, modName = model$name)

  model$rxout     = function(result)    {
    result        = as.data.frame(result)
    result = mutate(result,
                    Dtot1 = D1+DS1,
                    Stot1 = S1+DS1,
                    Dtot3 = D3+DS3,
                    Stot3 = S3+DS3,
                    Mtot1 = M1+DM1,
                    Mtot3 = M3+DM3)
  }

  return(model)
}
```

```
# Global Variables
model = ivsc_4cmtct_shedct()
tmax = 3*28 # End of the observation time, unit=day
tau = 14 # dosing interval, unit=day
```

```r
compartment = 2 # compartment to which dosing is applied

# Import parameters
d = xlsx::read.xlsx("../data/ModelF_Atezolizumab_Params.xlsx",1)
param.as.double = d$Value
names(param.as.double) = d$Parameter

# Function for ranges
lseq = function(from, to, length.out){
    sequence = seq(log(from), log(to), length.out=length.out)
    sequence = exp(sequence)
    return(sequence)
}
```

The function below calculate AFIRT from theory

```r
AFIRT_theory = function(dose.nmol){
    p = as.data.frame(t(param.as.double))
    Kss = with(p, (koff3 + keDM3 + kshedM3)/kon3)
    Kd = with(p, koff3 / kon3)

    # numerator and denomenator for Mtot3.ss(Mtot3 at steady state)
    numerator = with(p, k13DM*(VD1/VD3)*ksynM1+(keDM1+kshedM1+k13DM)*kshedM3)
    denomenator = with(p, (keDM1+kshedM1+k13DM)*(keDM3+kshedM3+k31DM)-k31DM*k13DM)
    Mtot3.ss = numerator / denomenator

    # numerator and denomenator for M3.0 (M3 at initial state)
    numerator = with(p, k13M*(VD1/VD3)*ksynM1+(keM1+kshedM1+k13D)*kshedM3)
    denomenator = with(p, (keM1+kshedM1+k13D)*(keD3+kshedM3+k31D)-k31D*k13D)
    M3.0 = numerator / denomenator

    # Target accumulation in the tumor compartment
    Tacc.tum = Mtot3.ss / M3.0

    CL = with(p, keD1 / VD1)
    B = with(p, (k13D*VD1/VD3)/(keD3 + k31D))

    AFIRT.theory.Kss = Kss*Tacc.tum*(CL*tau)/(dose.nmol*B)
    AFIRT.theory.Kd = Kd*Tacc.tum*(CL*tau)/(dose.nmol*B)
    return(c(AFIRT.theory.Kss, AFIRT.theory.Kd))
}
```

The function below simulates AFIRT as the average of free target to initial target

```r
AFIRT_sim = function(dose.nmol){
  ev = eventTable(amount.units="nmol", time.units="days")
  sample.points = c(seq(-7, tmax, 0.1), 10^(-3:0)) # sample time, increment by 0.1
  sample.points = sort(sample.points)
  sample.points = unique(sample.points)
  ev$add.sampling(sample.points)
  ev$add.dosing(dose=dose.nmol, nbr.doses=floor(tmax/tau)+1, dosing.interval=tau,
                dosing.to=2)

  init = model$init(param.as.double)
  out = model$rxode$solve(param.as.double, ev, init)
  out = model$rxout(out)
  out = out %>%
    mutate(Sfree.pct = S1/init["S1"],
             Mfree.pct = M3/init["M3"],
             dose.nmol = dose.nmol)

  last.two.doses = out %>%
    filter(time > 2*28 & time < 3*28)

  AFIRT.sim = mean(last.two.doses$Mfree.pct)

  return(AFIRT.sim)
}
```

The function below performs sensitivity analysis for AFIRT

```
sensitivity_analysis_for_AFIRT = function(dose.nmol, variable, range){
    df = data.frame()
    for (value in range){
        param.as.double[variable] = value
        AFIRT.sim = AFIRT_sim(dose.nmol)
        AFIRT.theory = AFIRT_theory(dose.nmol)
        row = append(c(value, AFIRT.sim), AFIRT.theory)
        df = rbind(df, row)
    }
    colnames(df) = c(variable, "AFIRF.sim", "AFIRT.theory.Kss", "AFIRT.theory.Kd")
    return(df)
}

# A function that plots the sensitivity analysis
plot.AFIRT.sensitivity.analysis = function(data, filename){
    names = names(data)
    data = data %>% gather(key, value, -c(get(names[1])))
    g = ggplot(data, aes(get(names[1]), value, color=key)) +
        scale.x.log10() +
        scale.y.log10() +
        geom_point() +
        ylab("AFIRT") +
        xlab(names[1])
    ggsave(filename, g)
    return(g)
}
```

Simulated output, initial dose = 80nmol, let's see if the model does what it supposes to do

```
simulation = function(dose.nmol){
  ev = eventTable(amount.units="nmol", time.units="days")
  sample.points = c(seq(-7, tmax, 0.1), 10^(-3:0)) # sample time, increment by 0.1
  sample.points = sort(sample.points)
  sample.points = unique(sample.points)
  ev$add.sampling(sample.points)
  ev$add.dosing(dose=dose.nmol, nbr.doses=floor(tmax/tau)+1, dosing.interval=tau,
                dosing.to=2)

  init = model$init(param.as.double)
  out = model$rxode$solve(param.as.double, ev, init)
  out = model$rxout(out)
  out = out %>%
    mutate(Sfree.pct = S1/init["S1"],
             Mfree.pct = M3/init["M3"],
             dose.nmol = dose.nmol)
  return(out)
}

dose.nmol = 80
df = simulation(dose.nmol)
```

Here is how the simulated output looks like for time > 0

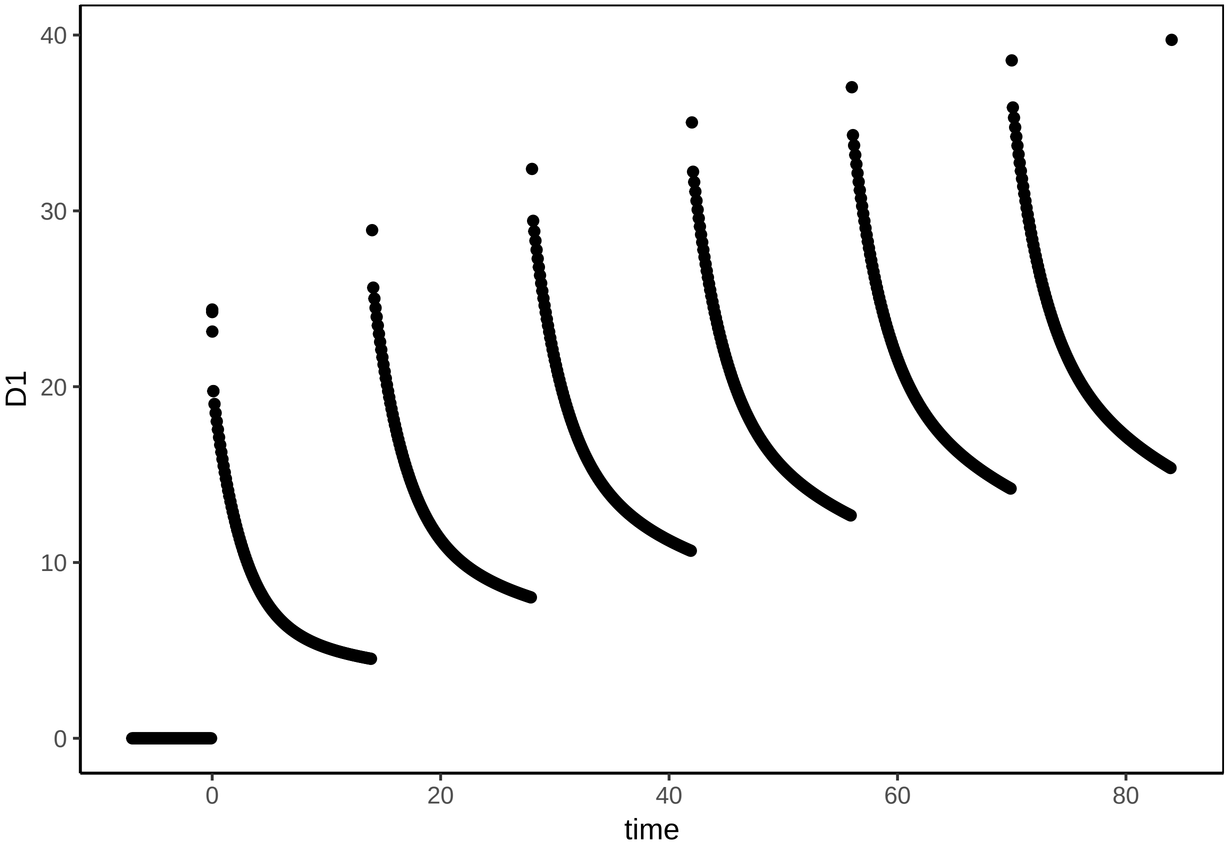
```
dose_applied = df %>%
  filter(time >0)
print(head(dose_applied, 5))
```

```
##    time AmtD0    AmtD1         D2          D3        S1        S3       M3
## 1 0.001     0 79.51844 0.00365723 0.003628997 4.4840537 10.148272 2.551297
## 2 0.010     0 75.88162 0.03565906 0.033063777 3.4877813 10.143403 2.548413
## 3 0.100     0 64.77432 0.31536073 0.179746574 0.7558130  9.934366 2.459079
## 4 0.100     0 64.77432 0.31536073 0.179746574 0.7558130  9.934366 2.459079
## 5 0.200     0 62.37178 0.59930354 0.260878983 0.5712689  9.625067 2.399031
##         DS1          DS3          DM3         DM1         M1       D1
## 1 0.1325826 3.220856e-05 3.380323e-05 0.008589817 0.29102625 24.24343
## 2 1.1283860 2.978358e-03 2.917720e-03 0.072067575 0.22754849 23.13464
## 3 3.9202573 1.652914e-01 9.225223e-02 0.229117185 0.07049886 19.74827
## 4 3.9202573 1.652914e-01 9.225223e-02 0.229117185 0.07049886 19.74827
## 5 4.1988663 4.245080e-01 1.523003e-01 0.236587658 0.06302838 19.01579
##      Dtot1    Stot1       Dtot3    Stot3     Mtot1    Mtot3  Sfree.pct
## 1 24.37601 4.616636 0.003661206 10.14830 0.2996161 2.551331 0.41956009
## 2 24.26303 4.616167 0.036042135 10.14638 0.2996161 2.551331 0.32634173
## 3 23.66853 4.676070 0.345037965 10.09966 0.2996160 2.551331 0.07071926
## 4 23.66853 4.676070 0.345037965 10.09966 0.2996160 2.551331 0.07071926
## 5 23.21465 4.770135 0.685386963 10.04958 0.2996160 2.551331 0.05345200
##   Mfree.pct dose.nmol
## 1  1.087546        80
## 2  1.086317        80
## 3  1.048236        80
## 4  1.048236        80
## 5  1.022640        80
```

There is something wrong with the model. We can see it by looking at the plot of some columns of the above data frame

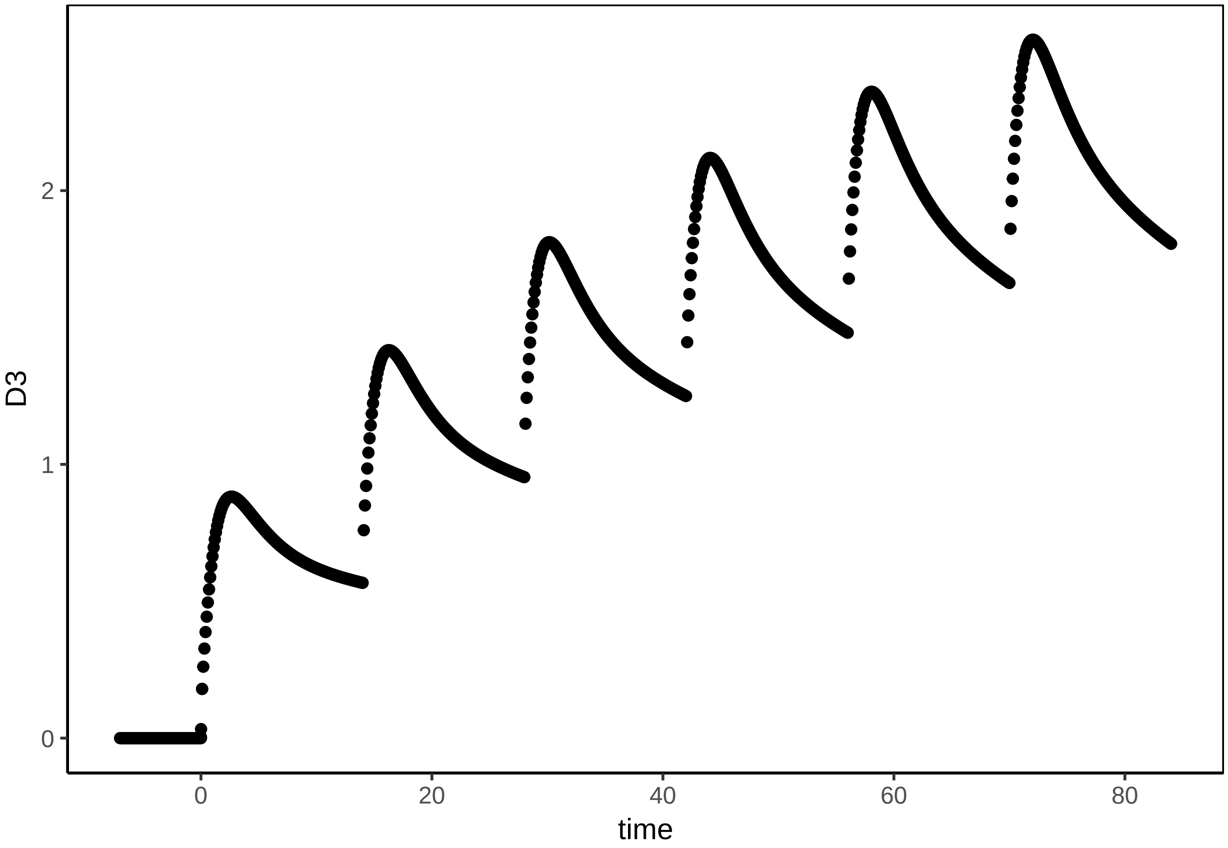
D1 agaist time

```
g = ggplot(df, aes(time, D1)) + geom_point()
print(g)
```



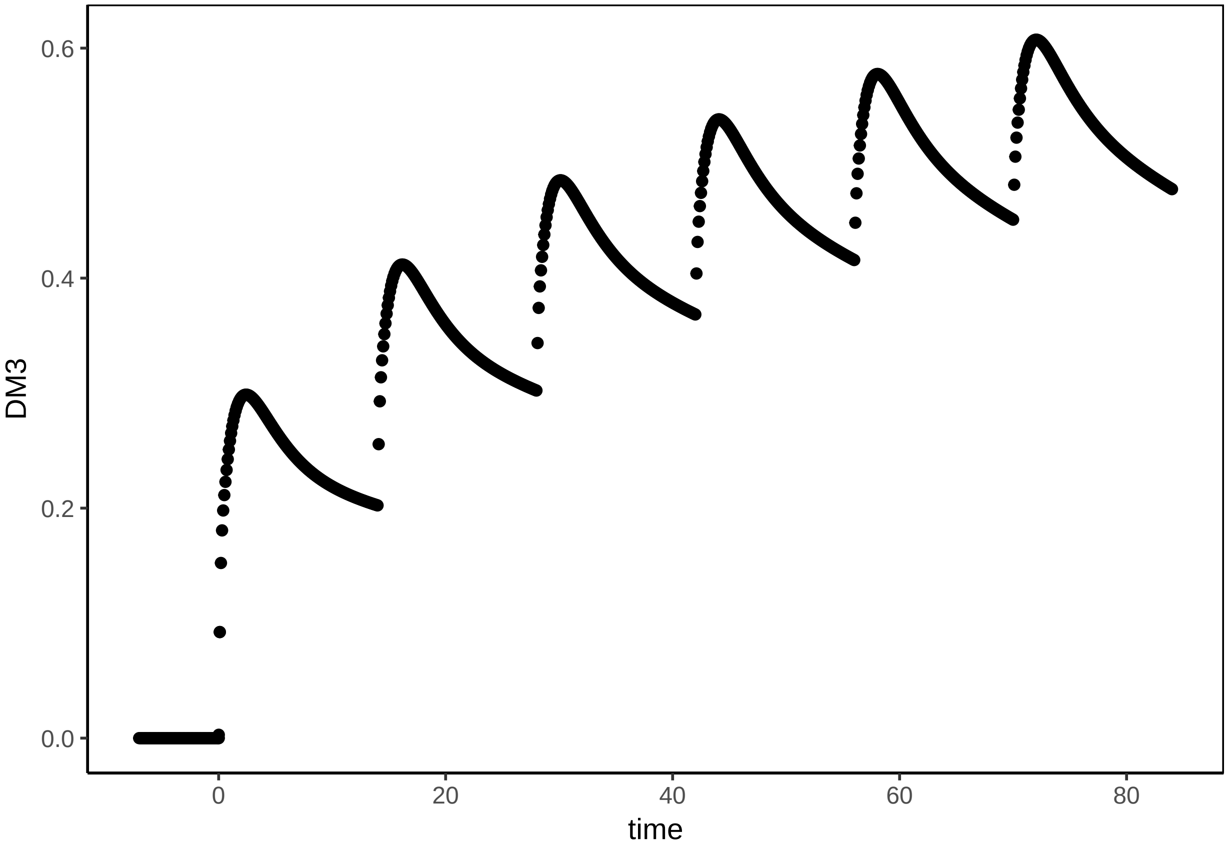
D3 agaist time

```
g = ggplot(df, aes(time, D3)) + geom_point()
print(g)
```



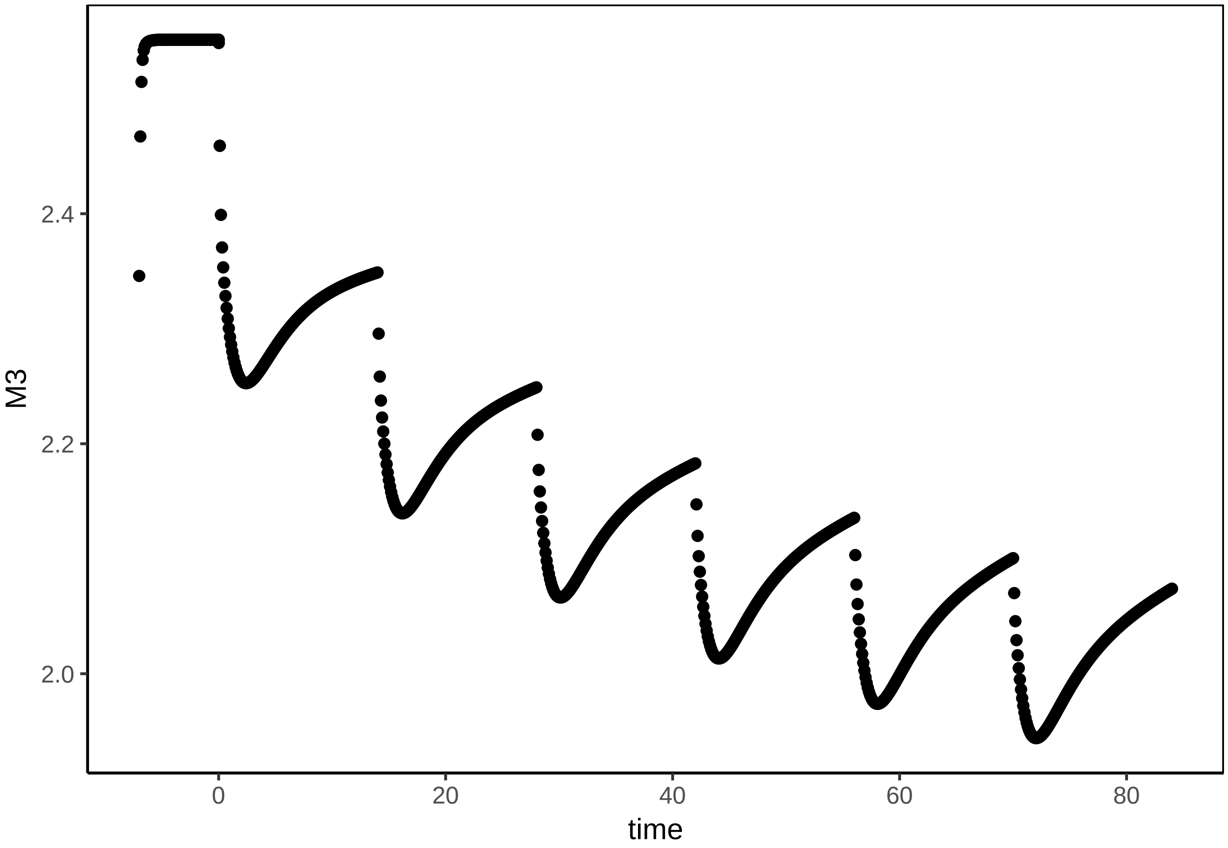
# complex against time

```
g = ggplot(df, aes(time, DM3)) + geom_point()
print(g)
```



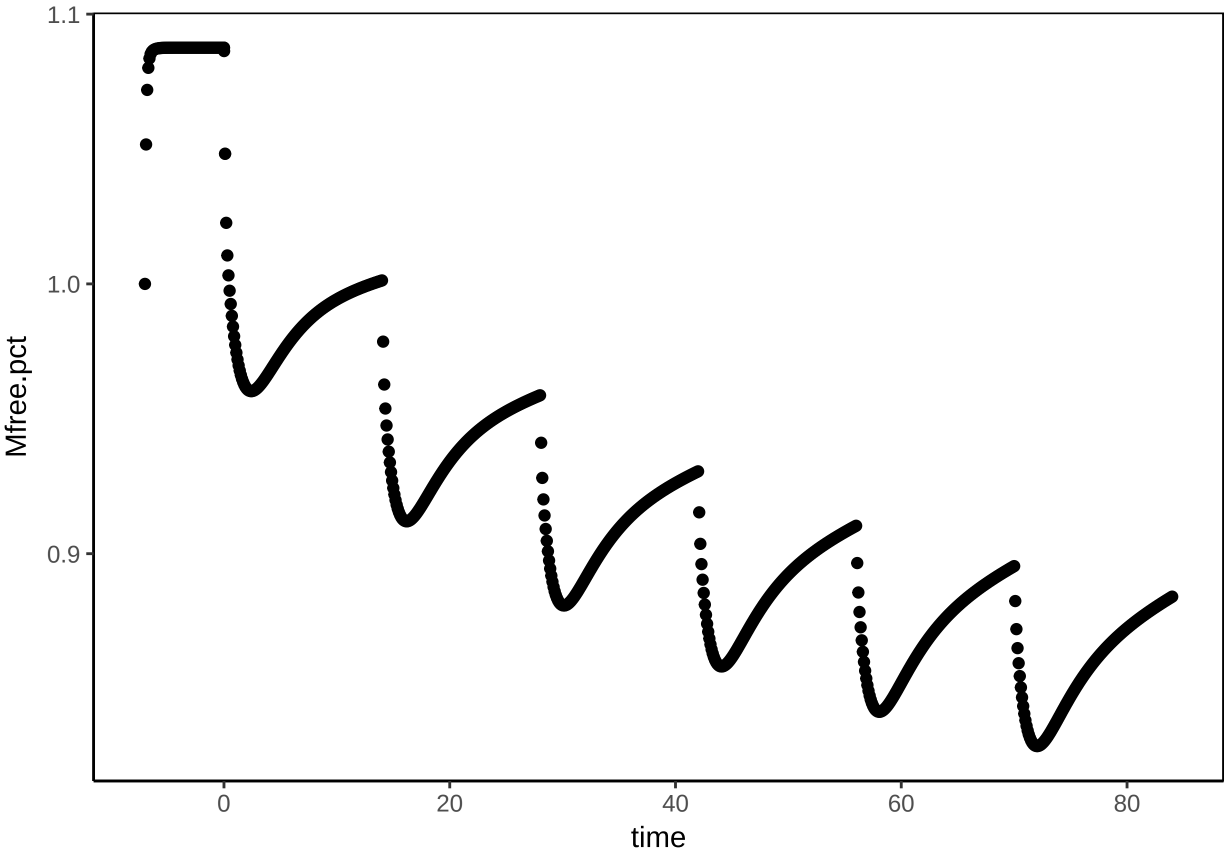
# Free target agaist time

```r
g = ggplot(df, aes(time, M3)) + geom_point()
print(g)
```



Free target to initial target agaist time

```
g = ggplot(df, aes(time, Mfree.pct)) + geom_point()
print(g)
```



Perform sensitivity analysis for AFIRT on dose.nmol

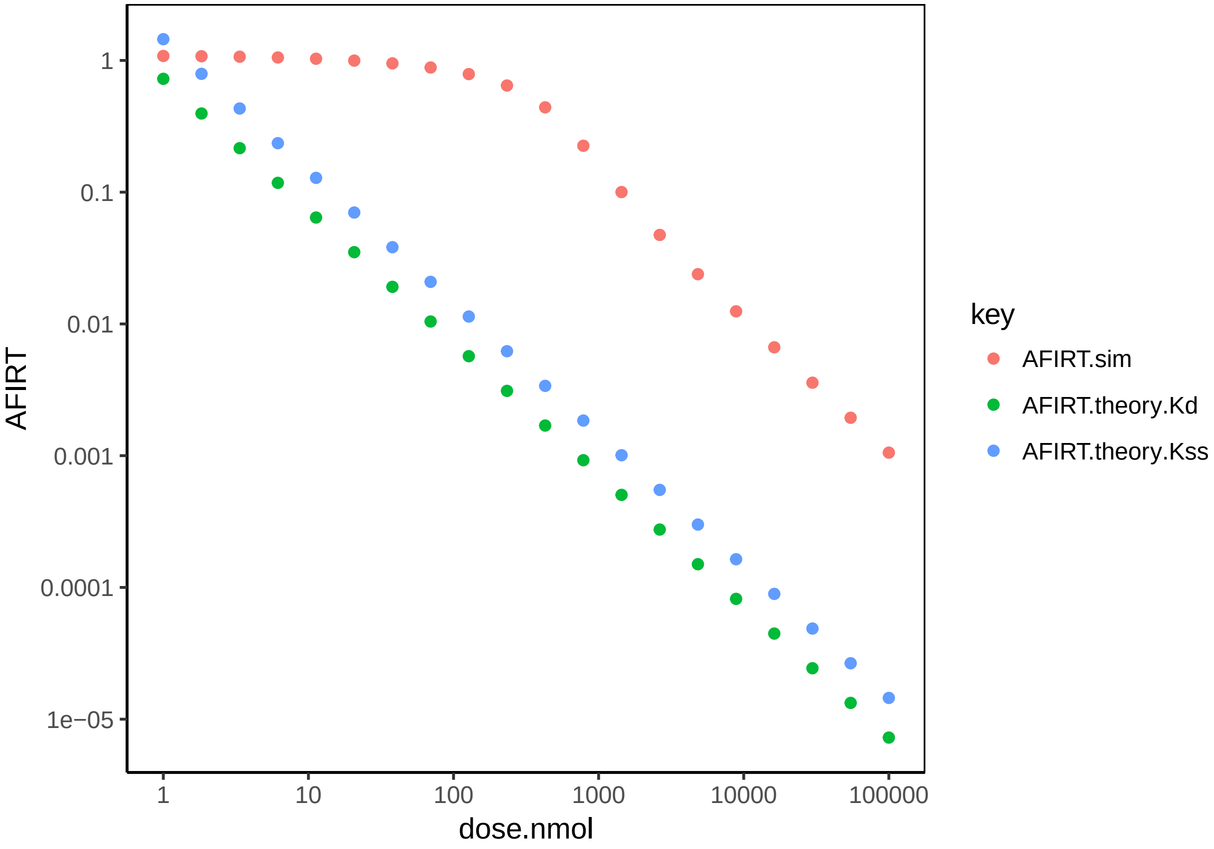
```
sensitivity_analysis_wrt_dose.nmol = function(dose.nmol.range){
  df = data.frame()
  for (dose.nmol in dose.nmol.range){
    AFIRT.sim = AFIRT_sim(dose.nmol=dose.nmol)
    AFIRT.theory = AFIRT_theory(dose.nmol = dose.nmol)
    row = append(c(dose.nmol, AFIRT.sim), AFIRT.theory)
    df = rbind(df, row)
  }
  colnames(df) = c("dose.nmol", "AFIRT.sim", "AFIRT.theory.Kss", "AFIRT.theory.Kd")
  return(df)
}

dose.nmol.range = lseq(1, 100000, 20)
df = sensitivity_analysis_wrt_dose.nmol(dose.nmol.range = dose.nmol.range)
print(df)
```

```
##       dose.nmol   AFIRT.sim AFIRT.theory.Kss AFIRT.theory.Kd
## 1  1.000000e+00 1.081043111     1.450083e+00    7.250417e-01
## 2  1.832981e+00 1.075903715     7.911067e-01    3.955534e-01
## 3  3.359818e+00 1.067081282     4.315958e-01    2.157979e-01
## 4  6.158482e+00 1.052566886     2.354612e-01    1.177306e-01
## 5  1.128838e+01 1.030007110     1.284581e-01    6.422903e-02
## 6  2.069138e+01 0.996926051     7.008152e-02    3.504076e-02
## 7  3.792690e+01 0.950077937     3.823364e-02    1.911682e-02
## 8  6.951928e+01 0.883661312     2.085872e-02    1.042936e-02
## 9  1.274275e+02 0.787511836     1.137967e-02    5.689837e-03
## 10 2.335721e+02 0.645021585     6.208289e-03    3.104144e-03
## 11 4.281332e+02 0.440595294     3.386991e-03    1.693495e-03
## 12 7.847600e+02 0.225093909     1.847805e-03    9.239025e-04
## 13 1.438450e+03 0.100244479     1.008087e-03    5.040437e-04
## 14 2.636651e+03 0.047370957     5.499717e-04    2.749858e-04
## 15 4.832930e+03 0.023859935     3.000423e-04    1.500211e-04
## 16 8.858668e+03 0.012468204     1.636909e-04    8.184545e-05
## 17 1.623777e+04 0.006645794     8.930312e-05    4.465156e-05
## 18 2.976351e+04 0.003580275     4.872017e-05    2.436008e-05
## 19 5.455595e+04 0.001939924     2.657975e-05    1.328987e-05
## 20 1.000000e+05 0.001054407     1.450083e-05    7.250417e-06
```

Make a plot of the above data frame

```
plot.AFIRT.sensitivity.analysis(df, "AFIRTwrtdose.jpg")
```

```
## Saving 6.5 x 4.5 in image
```



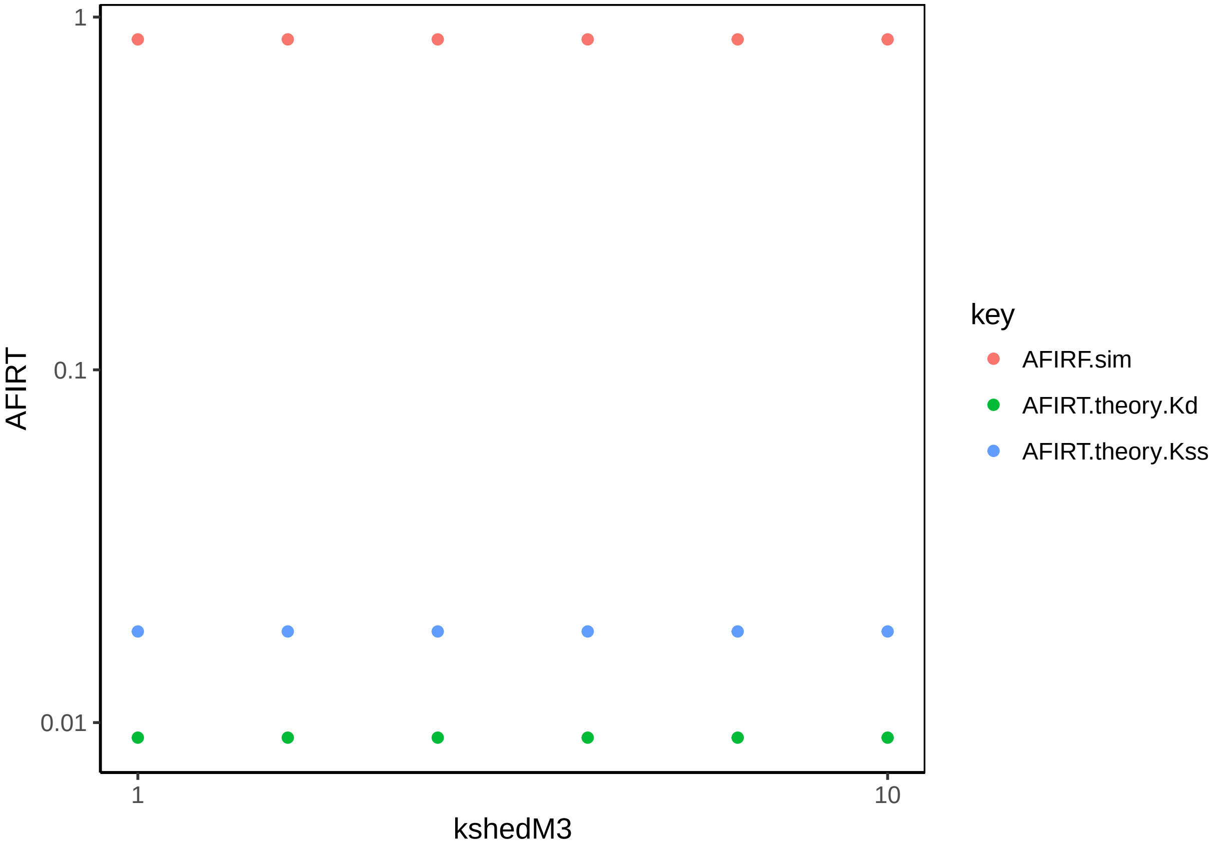
AFIRT sensitivity analysis on kshedM3

kshedM3 = 3 in the parameter file

set range of kshedM3 to be [1, 10] with 6 folds

```
sen = sensitivity_analysis_for_AFIRT(dose.nmol=80, variable="kshedM3", range = lseq(1, 10 ,6))
plot.AFIRT.sensitivity.analysis(sen, "AFIRTwrtkshedM3.jpg")
```

```
## Saving 6.5 x 4.5 in image
```



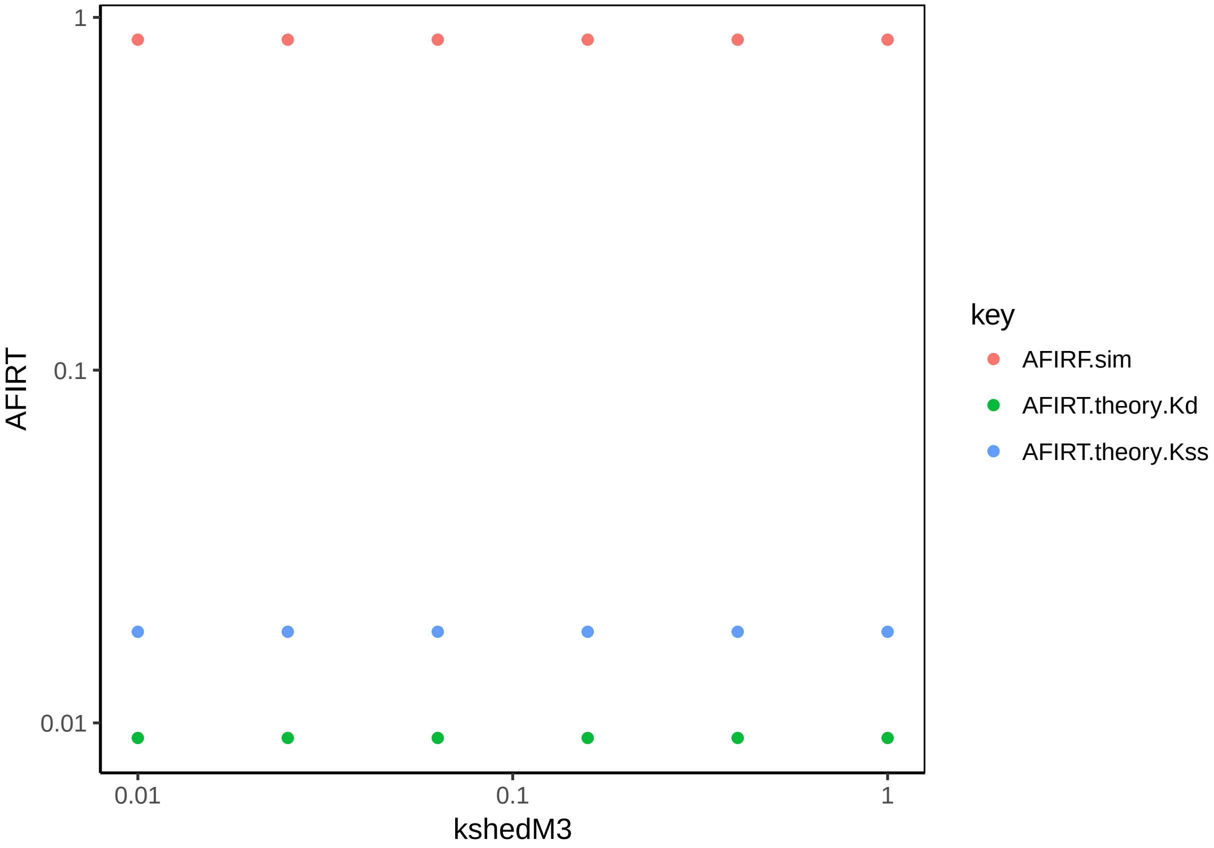
AFIRT sensitivity analysis on VD3 (tumor size)

VD3 = 0.1 in the parameter file

set range of VD3 to be [0.01, 1] with 6 folds

```
sen = sensitivity_analysis_for_AFIRT(dose.nmol=80, variable="kshedM3", range = lseq(0.01, 1, 6))
plot.AFIRT.sensitivity.analysis(sen, "AFIRTwrtVD3.jpg")
```

```
## Saving 6.5 x 4.5 in image
```


1
0.1
0.01
AFIRT
0.01
0.1
1
kshedM3
key
AFIRF.sim
AFIRT.theory.Kd
AFIRT.theory.Kss