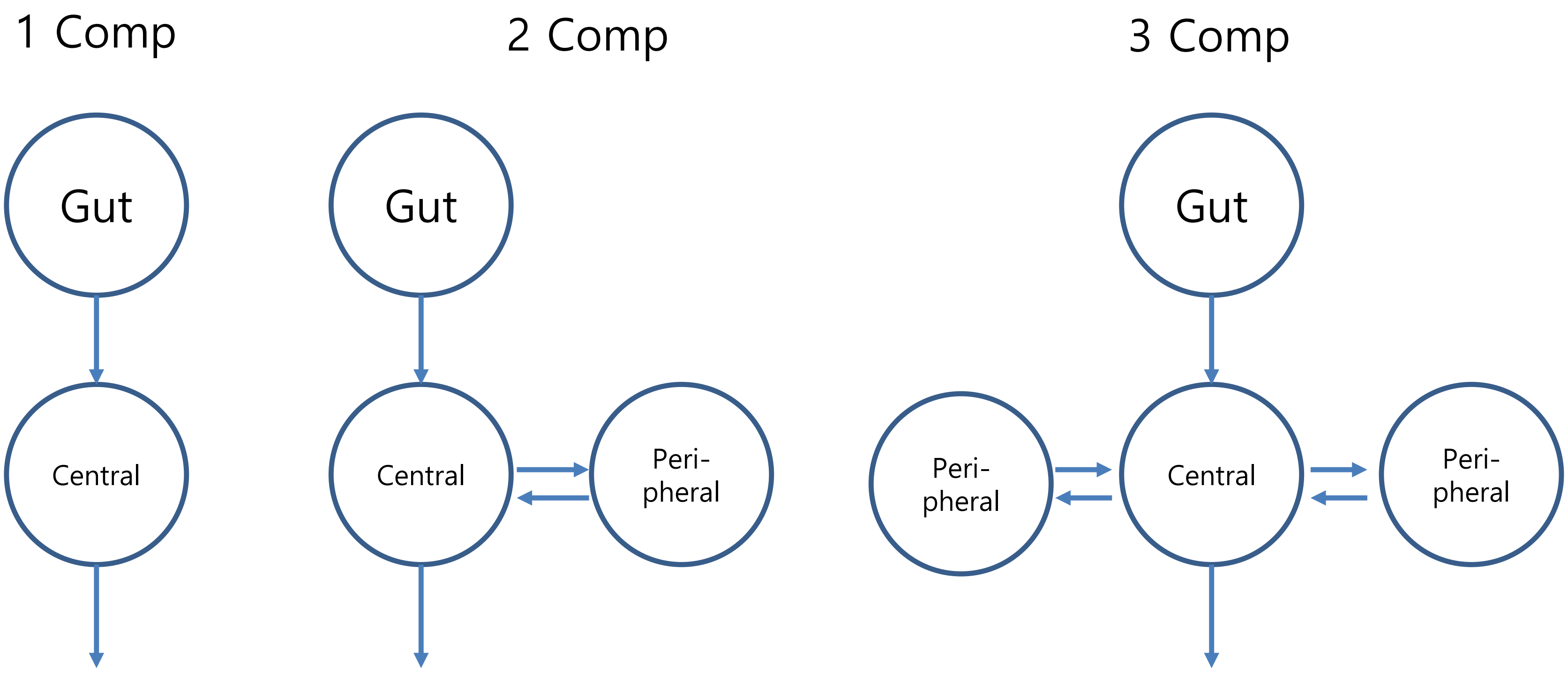
R을 이용한 PK-PD modeling

배 균 섭
서울아산병원
울산대학교
k@acr.kr

Compartment Model



Linear Coupled Differential Equation System

1 Comp

$$\frac{dX}{dt} = -K_e X + R + K_a X_g(0) e^{-K_a t}$$

2 Comp

$$\frac{d\vec{X}}{dt} = \begin{bmatrix} -(K_e + K_{cp}) & K_{pc} \\ K_{cp} & -K_{pc} \end{bmatrix} \vec{X} + \begin{bmatrix} R \\ 0 \end{bmatrix} + \begin{bmatrix} K_a X_g(0) e^{-K_a t} \\ 0 \end{bmatrix}$$

3 Comp

$$\frac{d\vec{X}}{dt} = \begin{bmatrix} -(K_{10} + K_{12} + K_{13}) & K_{21} & K_{31} \\ K_{12} & -K_{21} & 0 \\ K_{13} & 0 & -K_{31} \end{bmatrix} \vec{X} + \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} K_a X_g(0) e^{-K_a t} \\ 0 \\ 0 \end{bmatrix}$$

Reference

TCP Transl Clin Pharmacol

2019;27(2):43-51
https://doi.org/10.12793/tcp.2019.27.2.43

Analytical solution of linear multi-compartment models with non-zero initial condition and its implementation with R

David Z. D'Argenio[1] and Kyun-Seop Bae[2]*

[1]University of Southern California School of Engineering, [2]Asan Medical Center, University of Ulsan College of Medicine, Seoul 05505, Korea
*Correspondence: K. S. Bae; Tel: +82-2-3010-4611, Fax: +82-2-3010-4623, E-mail: ksbae@amc.seoul.kr

Principle for Solution

$$\frac{d\vec{X}}{dt} = A\vec{X} + \overrightarrow{f(t)} \text{ with the initial condition } \overrightarrow{X_o}$$

$$\overrightarrow{X(t)} = \overrightarrow{X_H(t)} + \overrightarrow{X_P(t)}$$

$$\overrightarrow{X(t)} = \exp(At)\,\overrightarrow{X_o} + \exp(At) * \overrightarrow{f(t)}$$

$$\overrightarrow{X(t)} = \exp(At)\,\overrightarrow{X_o} + \int_0^t \exp[A(t-\tau)]\,f(\tau)d\tau$$

Case Presentation

1. Dosing History

– 0 hr, 100 mg IV bolus.

– 24 hr, 150 mg IV infusion with the rate of 50mg/hr

– 48 hr, 100 mg PO

2. Observation Time Points

– 1, 2, 4, 8, 12 hour after each dosing

Simulation Scenario

1. **One-compartment model**

   Ka=1, Ke=0.1, F=1, V=1

2. **Two-compartment model**

   Ka=1, Ke=K10=0.1, K12=3, K21=1, F=1, V=1,

3. **Three-compartment model**

   Ka=1, Ke=K10=0.1, K12=3, K21=1, K13=2, K31=0.5, F=1, V=1

Data Prep

```
install.packages("wnl") ; require(wnl)

TIME0 = c(1, 2, 4, 8, 12)
TIME = c(TIME0, TIME0 + 24, TIME0 + 48)
DV = rep(NA, length(TIME))
Obs = cbind(TIME, DV)

TIME = c(0, 24, 48)
AMT = c(100, 150, 100)
RATE = c(0, 50, 0)
CMT = c(2, 2, 1)
DoseHist = cbind(TIME, AMT, RATE, CMT)

DAT = merge(Obs, DoseHist, all=TRUE) ; DAT
DAT2 = ExpandDH(DAT) ; DAT2
```

R script

```
# 1 Comp
X1 = Comp1(Ke=0.1, Ka=1, DAT2) ; X1
matplot(DAT2[, "TIME"], X1, type="l")

# 2 Comp
Sol = SolComp2(K10=0.1, K12=3, K21=1)
X2 = nComp(Sol, Ka=1, DAT2) ; X2
matplot(DAT2[, "TIME"], X2, type="l")

# 3 Comp
Sol = SolComp3(K10=0.1, K12=3, K21=1, K13=2, K31=0.5)
X3 = nComp(Sol, Ka=1, DAT2) ; X3
matplot(DAT2[, "TIME"], X3, type="l")
```

Result – Data Prep

 # DAT

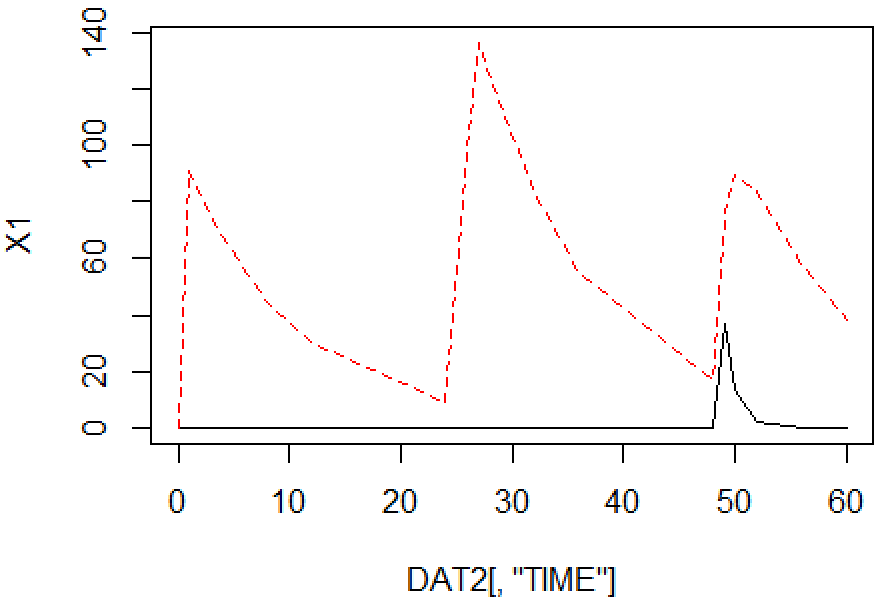
	TIME	AMT	RATE	CMT	DV
1	0	100	0	2	NA
2	1	NA	NA	NA	NA
3	2	NA	NA	NA	NA
4	4	NA	NA	NA	NA
5	8	NA	NA	NA	NA
6	12	NA	NA	NA	NA
7	24	150	50	2	NA
8	25	NA	NA	NA	NA
9	26	NA	NA	NA	NA
10	28	NA	NA	NA	NA
11	32	NA	NA	NA	NA
12	36	NA	NA	NA	NA
13	48	100	0	1	NA
14	49	NA	NA	NA	NA
15	50	NA	NA	NA	NA
16	52	NA	NA	NA	NA
17	56	NA	NA	NA	NA
18	60	NA	NA	NA	NA

 # DAT2

	TIME	AMT	RATE	CMT	DV	BOLUS	RATE2
1	0	100	0	2	0	100	0
2	1	0	0	0	0	0	0
3	2	0	0	0	0	0	0
4	4	0	0	0	0	0	0
5	8	0	0	0	0	0	0
6	12	0	0	0	0	0	0
7	24	150	50	2	0	0	50
8	25	0	0	2	0	0	50
9	26	0	0	2	0	0	50
19	27	0	0	0	0	0	0
10	28	0	0	0	0	0	0
11	32	0	0	0	0	0	0
12	36	0	0	0	0	0	0
13	48	100	0	1	0	100	0
14	49	0	0	0	0	0	0
15	50	0	0	0	0	0	0
16	52	0	0	0	0	0	0
17	56	0	0	0	0	0	0
18	60	0	0	0	0	0	0

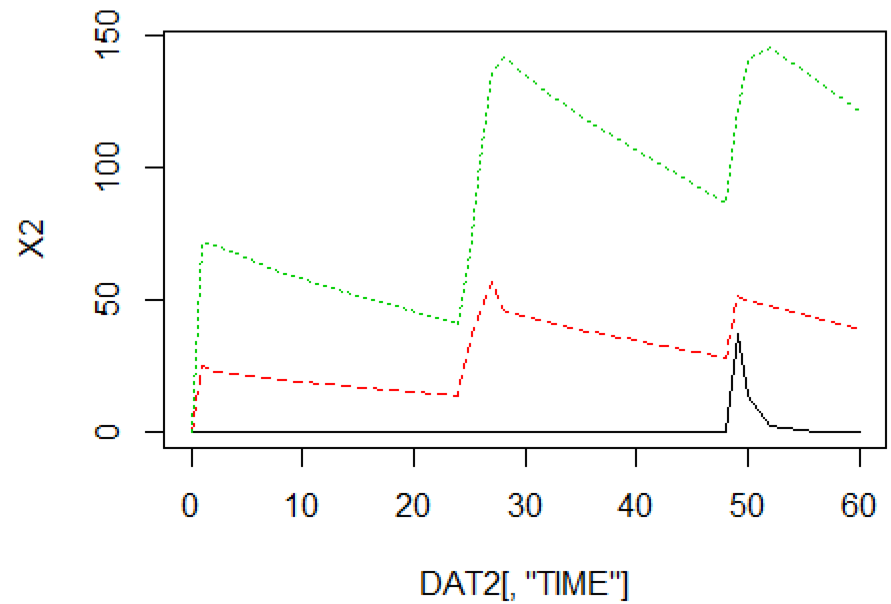
Result – 1 Comp

```
             [,1]       [,2]
 [1,] 0.000000e+00   0.000000
 [2,] 0.000000e+00  90.483742
 [3,] 0.000000e+00  81.873075
 [4,] 0.000000e+00  67.032005
 [5,] 0.000000e+00  44.932896
 [6,] 0.000000e+00  30.119421
 [7,] 0.000000e+00   9.071795
 [8,] 0.000000e+00  55.789791
 [9,] 0.000000e+00  98.061981
[10,] 0.000000e+00 136.311441
[11,] 0.000000e+00 123.339692
[12,] 0.000000e+00  82.677068
[13,] 0.000000e+00  55.420096
[14,] 0.000000e+00  16.692212
[15,] 3.678794e+01  74.765736
[16,] 1.353353e+01  89.599257
[17,] 1.831564e+00  83.634059
[18,] 3.354626e-02  57.388461
[19,] 6.144212e-04  38.492939
```



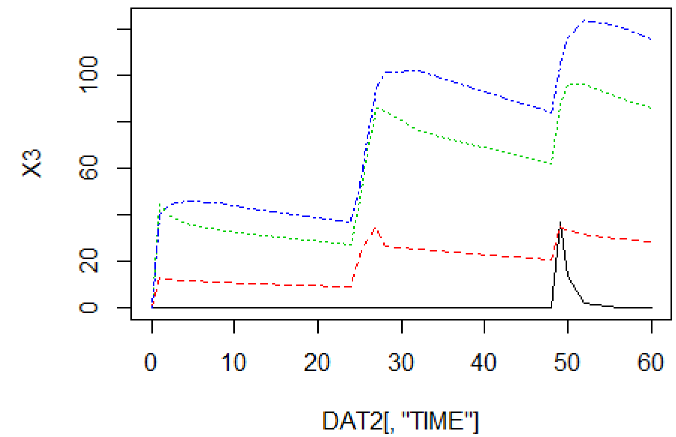
Result – 2 Comp

	[,1]	[,2]	[,3]
[1,]	0.000000e+00	0.00000	0.00000
[2,]	0.000000e+00	24.78579	71.00430
[3,]	0.000000e+00	22.94872	70.48922
[4,]	0.000000e+00	21.82887	67.13386
[5,]	0.000000e+00	19.78818	60.85782
[6,]	0.000000e+00	17.93827	55.16849
[7,]	0.000000e+00	13.36302	41.09748
[8,]	0.000000e+00	34.08869	67.74790
[9,]	0.000000e+00	45.53336	102.31559
[10,]	0.000000e+00	56.55153	136.19086
[11,]	0.000000e+00	46.25053	141.60095
[12,]	0.000000e+00	41.78336	128.50319
[13,]	0.000000e+00	37.87722	116.48999
[14,]	0.000000e+00	28.21644	86.77862
[15,]	3.678794e+01	51.20062	122.37609
[16,]	1.353353e+01	50.36158	141.38048
[17,]	1.831564e+00	47.95656	145.65708
[18,]	3.354626e-02	43.47332	133.66703
[19,]	6.144212e-04	39.40919	121.20089



Result – 3 Comp

	[,1]	[,2]	[,3]	[,4]
[1,]	0.000000e+00	0.000000	0.00000	0.00000
[2,]	0.000000e+00	13.004819	44.19187	40.02386
[3,]	0.000000e+00	12.205766	39.85761	43.90691
[4,]	0.000000e+00	11.604779	35.94874	46.04289
[5,]	0.000000e+00	10.954285	33.31609	44.82325
[6,]	0.000000e+00	10.422939	31.66063	42.73598
[7,]	0.000000e+00	8.993606	27.31648	36.88080
[8,]	0.000000e+00	22.781008	46.57865	51.99548
[9,]	0.000000e+00	28.923842	67.17024	72.67283
[10,]	0.000000e+00	34.824944	86.10027	94.65280
[11,]	0.000000e+00	26.682954	84.48999	101.57568
[12,]	0.000000e+00	24.912956	75.89143	101.67155
[13,]	0.000000e+00	23.688228	71.96288	97.10912
[14,]	0.000000e+00	20.438779	62.07916	83.81492
[15,]	3.678794e+01	34.919791	87.37083	103.91654
[16,]	1.353353e+01	33.228471	96.38124	116.45056
[17,]	1.831564e+00	31.441383	95.92774	123.94904
[18,]	3.354626e-02	29.630545	90.10716	121.19273
[19,]	6.144212e-04	28.190201	85.63118	115.58182



5th edition

Pharmacokinetic and Pharmacodynamic Data Analysis

Concepts and Applications

Johan Gabrielsson
Dan Weiner

Apotekarsocieteten

Exercise	Model	Route	Input	Acute/ chronic	Linear or nonlinear	Analyte	Simul fitting	Large / Small molecule	DE or Closed
PK1	1cmpt	IV	Bolus	Single	L	Plasma	N	Small	both
PK2	1cmpt	Oral	1st	Single	L	Plasma	N	Small	Closed
PK3	1cmpt	Oral	1st and 0	Single	L	Plasma	N	Small	both
PK4	1cmpt	Oral	1st	Single	L	Plasma	N	Small	Closed
PK5	1cmpt	IV	Single	Single	L	Plasma/Urine	Y	Small	both
PK6	1cmpt	IV	Single	Single	L	Plasma/Urine	Y	Small	both
PK7	1-3cmpt	IV	Bolus	Single	L	Plasma	N	Small	both
PK8	Distribution	IV	Bolus	Single	L	Plasma	N	Small	both
PK9	2cmpt	IV7PO	Bolus, 1st	Single	NL/Capacity	Plasma	N	Small	DE
PK10	2cmpt	IV/PO	Bolus, 1st	Single	L	Plasma	Y	Small	Closed
PK11	2cmpt	Oral	1st	Multiple	L	Plasma/Effect	N	Small	Closed
PK12	2cmpt	IV/PO	Bolus, 1st	Single	L	Plasma	Y	Small	Closed
PK13	2cmpt	IV	Bolus, infus	Single	L	Plasma	N	Small	both
PK14	2cmpt	Oral	1st	Single	L	Plasma	Y	Small	Closed
PK15	1cmpt	Oral	1st	Multiple	L	Plasma/Effect	N	Small	NCA
PK16	2cmpt	IV	Infusions	Acute	L	Plasma/Urine	Y	Small	DE
PK17	1cmpt	IV	Infusion	Single	N/Capacity	Plasma	N	Small	DE
PK18	2cmpt	IV	Infusion	Single	N/Capacity	Plasma	Y	Small	DE
PK19	2cmpt	IV	Multiple	Single	N/Capacity	Parent/Met	Y	Small	DE
PK20	1cmpt	IV	Bolus	Acute	N/Capacity	Plasma	Y	Small	DE
PK21	1cmpt	Oral	1st	Multiple	N/Time	Plasma	N	Small	DE
PK22	1cmpt	IV	Infusions	Multiple	N/Time	Plasma	N	Small	DE
PK23	Hepatic		Infusion		L	Plasma	N	Small	Closed
PK24	3cmpt	IV	Infusion	Single	N/Flow	Plasma	N	Small	DE
PK25	2cmpt	IV	Bolus	Single	L	Plasma/Urine	Y	Small	DE

Exercise	Model	Route	Input	Acute/ chronic	Linear or nonlinear	Analyte	Simul fitting	Large / Small molecule	DE or Closed
PK26	2cmpt	IV	Bolus	Acute	N/Capacity	Plasma	N	Large	DE
PK27	TMDD	IV	Bolus	Acute	N/TMDD	Plasma	Y	Large	
PK28	1cmpt	IV	Bolus	Single	L/Allometry	Plasma	Y	Small	Closed
PK29	2cmpt	IV	Bolus	Single	L/Allometry	Plasma	Y	Small	Closed
PK30	Turnover	IV	Infusion	Single	L	Plasma	N	Small	DE
PK31	Turnover	IV	Bolus/Inf	Single	L	Plasma	N	Small	DE
PK32	Turnover	IV	Bolus/Inf	Multiple	N	Plasma	N	Small	DE
PK33	1cmpt	TD	0-order	Single	L	Plasma	N	Small	DE
PK34	Reversible	IV	Infusion	Single	L	Plasma	Y	Small	DE
PK35	1cmpt	IV	Bolus	Single	L	Plasma	N	Small	Closed
PK36	1cmpt	SC	1^{st}	Multiple	L/oscillatory	Plasma	N	Small	DE
PK37	Hepatic	*In vitro*			L	*In vitro*	N	Small	Closed
PK38	Hepatic	*In vitro*			N	*In vitro*	N	Small	Closed
PK39	2cmpt	IV	Infusions	Multiple	L	Plasma	N	Small	DE
PK40	2cmpt EHP	IV	Bolus	Single	L	Plasma	Y	Small	DE
PK41	1cmpt	IV	Infusion	Single	N	Plasma	Y	Small	DE
PK42	2cmpt	Oral	1^{st}	Single	N	Plasma	N	Small	DE
PK43	1cmpt	Oral	1^{st}	Single	L	Plasma	N	Small	DE
PK44	Metabolic	*In vitro*			N	*In vitro*	N	Small	Closed
PK45	Reversible	IV	Infusion	Single	L	Plasma	Y	Small	DE
PK46	1cmpt	IV	Infusion	Single	L	Plasma	N	Small	ClosedNCA
PK47	Binding	*In vitro*			N/Binding	*In vitro*	Y	Small	Closed
PK48	1cmpt/ metab	IV	Bolus	Single	N	Plasma/Urine	Y	Small	DE
PK49	Metabolic	*In vitro*			N	*In vitro*	Y	Small	DE
PK50	2cmpt	IV	Infusion	Single	L	Plasma/Effect	N	Small	Closed
PK51	2cmpt met	IV/PO	Bolus/1^{st}	Single	L	Plasma	Y	Small	DE
PK52	2cmpt	IV	Bolus	Single	L	Plasma	N	Large	DE
PK53	2cmpt	IV	Infusions	Multile	L	Plasma	Y	Large	Closed

R script for PK13

```
setwd("C:/G/Rt/Gab/") ; require(wnl)
dPK13 = read.csv("PK13.csv", skip=1) ; colnames(dPK13) = c("TIME", "DV") ; dPK13

TIME = c(0, 0) ; AMT = c(400, 800) ; RATE = c(0, 800/26) ; CMT = c(2, 2)
DH = cbind(TIME, AMT, RATE, CMT)

dPK13a = merge(dPK13, DH, all=TRUE) ; dPK13a
dPK13b = ExpandDH(dPK13a) ; dPK13b

fPK13 = function(THETA)
{
  Vc  = THETA[1]
  K10 = THETA[2]
  K12 = THETA[3]
  K21 = THETA[4]
  Sol = SolComp2(K10, K12, K21)
  X1 = nComp(Sol, Ka=0, dPK13b)
  X1[,2] = X1[,2]/Vc
  return(X1[dPK13b[,"TIME"] %in% dPK13[,"TIME"], 2])
}
# Test fPK13
Vc=2.917; K10=0.1183; K12=0.0601; K21=0.0808 ; fPK13(c(Vc, K10, K12, K21))

nlr(fPK13, dPK13, pNames=c("Vc", "K10", "K12", "K21"), IE=c(2, 0.2, 0.1, 0.05),
    Error="C", SecNames=c("Vt", "Cl", "Cld"), SecForms=c(~Vc*K12/K21, ~Vc*K10, ~Vc*K12))
```

R Script Part 1 for PK12

```
require(wnl) ; require(deSolve)
dPK12 = read.csv("PK12.csv", skip=1) ; colnames(dPK12) = c("TIME", "DV") ; dPK12
Dpo = 2500

PKde = function(t, y, p)
{
  if (t >= 60 & t < 75 ) RateIn = 33.33333 # 500/15
  else RateIn = 0

  if (t < p["Tlag"]) Ka = 0
  else Ka = p["Ka"]

  dy1dt = -Ka*y[1]                                   # gut
  dy2dt = (RateIn + Ka*y[1] - p["Cl"]*y[2] - p["Cld"]*y[2] + p["Cld"]*y[3])/p["Vc"]
  dy3dt = (p["Cld"]*y[2] - p["Cld"]*y[3])/p["Vt"] # peripheral
  return(list(c(dy1dt, dy2dt, dy3dt)))
}
# Test PKde
Times = c(0, dPK12[,"TIME"])
iTime = 2:length(Times)
y = lsoda(y=c(0.0464*Dpo, 0, 0), times=Times, func=PKde,
          parms=c(Ka=0.104, Vc=0.12, Vt=0.2758, Cl=0.014566, Cld=0.02, Tlag=4.8694))
plot(dPK12[,"TIME"], dPK12[,"DV"]) ; lines(y[,1], y[,3])
```

R Script Part 2 for PK12

```
fPK12 = function(THETA)
{
  Fa   = THETA[1]
  Ka   = THETA[2]
  Vc   = THETA[3]
  Vt   = THETA[4]
  Cl   = THETA[5]
  Cld  = THETA[6]
  Tlag = THETA[7]

  Fs = lsoda(y=c(Fa*Dpo, 0, 0), times=Times, func=PKde,
             parms=c(Ka=Ka, Vc=Vc, Vt=Vt, Cl=Cl, Cld=Cld, Tlag=Tlag))
  return(Fs[iTime,"2"])
}

# Test fPK12
fPK12(c(0.0464, 0.104, 0.12, 0.2758, 0.014566, 0.02, 4.8694))

# Estimation
nlr(fPK12, dPK12, pNames=c("F", "Ka", "Vc", "Vt", "Cl", "Cld", "Tlag"),
    IE=c(0.05, 0.1, 0.2, 0.6, 0.02, 0.01, 2),
    LB=c(0.01, 0.01, 0.02, 0.06, 0.002, 0.001, 0),
    UB=c(1, 1, 100, 1000, 10, 100, 5))
```

Summary

- Necessary Packages
  - wnl
  - deSolve
- Usage of solution
  - Simulation
  - Estimation: Gabrielsson type, TDM