# Jupyter WebでR　１年生薬学概論用

# Jupyter Webにアクセス

# https://jupyter.org/try

# “Try Jupyter with R”をクリック、しばらく待つ

# File → New Notebook → R

# Rを電卓のように使ってみる

1 + 2 # Shift+Return

2 \* 3

2 ^ 4 # 24

2 ^ 50 # 250

sqrt(2)

log(10) # ln10

log10(100) # log100

# Rで統計

a <- c(2, 3, 4, 5, 6, 7, 8)

a

a <- a \* 10

a

sum(a)

mean(a)

sd(a)

b <- c(47, 48, 49, 50, 51, 52, 53)

b

mean(b)

sd(b)

boxplot(a, b)

# 1-コンパートメントモデルに基づく解析　経口投与　Jupyter用

time1 <- c(1, 2, 3, 4, 6, 8, 12)

conc1 <- c(22.3, 17.2, 15.1, 12.6, 5.7, 3.6, 1.2, 6.6, 10.3, 14.8, 10.4, 8.5, 7.1, 2.4)

eqid1 <- c(1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2)

dat <- data.frame(time1, conc1, eqid1)

names(dat)<-c('Time', 'Conc', 'Eqid')

print(dat)

plot(dat[,1], dat[,2], pch=ifelse(dat[,3]==1,17,1), cex=2, xlim=c(0,12), ylim=c(0,30), xlab='Time', ylab='Conc')

install.packages("minpack.lm")

library(minpack.lm)

# C = Dose/Vd \* exp(–(CL/Vd)\*t) 急速静注のとき

# C = (ka•F•Dose/(CL–ka•Vd))\*{exp(–ka\*t)–exp(–(CL/Vd)\*t)} 経口投与のとき

# p[1] = ka, p[2] = F, p[3] = CL, p[4] = Vd, Dose = 100

func <- function(p, x, flag){

y1=100/p[4]\*exp(-p[3]/p[4]\*x)

y2=100\*p[1]\*p[2]/(p[3]-p[1]\*p[4])\*(exp(-p[1]\*x)-exp(-p[3]/p[4]\*x))

c <- cbind(y1, y2)

diag(c[,flag])

}

res <- function(p, data){

data[,2]-func(p, data[,1], data[,3])

}

optimize <- function(p, data){

nls.lm(par=p, fn=res, data=data, control=nls.lm.control(maxiter=1000))

}

param <- c(0.8, 0.5, 0.3, 2)

result <- optimize(param, dat)

print(result)

func(coef(result), dat[,1], dat[,3])

plot(dat[,1], dat[,2], pch=ifelse(dat[,3]==1,17,1), cex=2, xlim=c(0,12), ylim=c(0,30), xlab='Time', ylab='Conc')

time <- seq(0, 12, 0.2)

lines(time, func(coef(result), time, rep(1,length(time))))

lines(time, func(coef(result), time, rep(2,length(time))))

# RでPK解析Simulation

# mrgsolveパッケージをインストール

install.packages("mrgsolve")

library(mrgsolve)

# 1-コンパートメントモデル経口投与

code<-"

$PARAM

Ke = 0.15, Vd = 10, Ka = 0.8

$INIT

Xa=10, X=0

$ODE

dxdt\_Xa = -Ka\*Xa;

dxdt\_X = Ka\*Xa - Ke\*X;

$CAPTURE

C = X/Vd; "

# グラフを描く

mod<-mcode("oral", code) %>% update(end = 24, delta = 0.1)

mod %>% mrgsim %>% plot

# Kaを変化させてみる

idataka <- expand.idata(Ka = c(0.4, 0.8, 1.6))

out <-

mod %>%

idata\_set(idataka) %>%

mrgsim(end=24)

plot(out, C~time)

# Vdを変化させてみる

idatavd <- expand.idata(Vd = c(5, 10, 20))

out <-

mod %>%

idata\_set(idatavd) %>%

mrgsim(end=24)

plot(out, C~time)

# Keを変化させてみる

idatake <- expand.idata(Ke = c(0.075, 0.15, 0.3))

out <-

mod %>%

idata\_set(idatake) %>%

mrgsim(end=24)

plot(out, C~time)

# Keが異なる患者へ12時間ごと10 mg繰り返し投与3日間

idatake <- expand.idata(Ke = c(0.01, 0.05, 0.25))

mod %>% init(Xa=0) %>%

ev\_rx("10 q 12 x 6") %>%

idata\_set(idatake) %>%

mrgsim(end = 72, delta = 0.1) %>%

plot(C~time)

# Keが異なる患者へ24時間ごと4 mg繰り返し投与30日間

idatake <- expand.idata(Ke = c(0.01, 0.05, 0.25))

mod %>% init(Xa=0) %>%

ev\_rx("4 q 24 x 30") %>%

idata\_set(idatake) %>%

mrgsim(end = 720, delta = 0.1) %>%

plot(C~time)

# Ka、Ke、Vdなどの数字を変えて試してみてください

# 終了するときは、ブラウザ（のタブ）を閉じてください。何も保存されません

# 保存したいときは、次のようにしてください

# File → Rename → 「R-PK」など名前をつける

# File → Download as → Notebook　としてダウンロード

# 便利な使い方

# グーグルコラボにアクセスする

# https://colab.research.google.com

# グーグルのアカウント（Gmailアドレスとパスワード）でログインする

# ファイル→ファイルをアップロード　先ほどダウンロードしたファイルをアップロード

# グーグルコラボ上で実行でき、保存もできます