Global Sensitivity Analysis on Voriconazole model

August 2020

```
using Pumas, Plots
```

0.1 Introduction

In this tutorial, we will cover running global sensitivity analysis on the Voriconazole model published here https://github.com/metrumresearchgroup/Voriconazole-PBPK/

0.1.1 Model Code

```
model = @model begin
    @param begin
        Fup ∈ RealDomain(init = 0.42)
        fumic ∈ RealDomain(init = 0.711)
        WEIGHT ∈ RealDomain(init = 73)
        MPPGL ∈ RealDomain(init = 30.3)
        MPPGI ∈ RealDomain(init = 0)
        C_OUTPUT ∈ RealDomain(init = 6.5)
        VmaxH ∈ RealDomain(init = 40)
        VmaxG ∈ RealDomain(init = 40)
        KmH ∈ RealDomain(init = 9.3)
        KmG ∈ RealDomain(init = 9.3)
        bp ∈ RealDomain(init = 1)
        kpad ∈ RealDomain(init = 9.89)
        kpbo ∈ RealDomain(init = 7.91)
        kpbr ∈ RealDomain(init = 7.35)
        kpgu ∈ RealDomain(init = 5.82)
        kphe ∈ RealDomain(init = 1.95)
        kpki ∈ RealDomain(init = 2.9)
        kpli ∈ RealDomain(init = 4.66)
        kplu ∈ RealDomain(init = 0.83)
        kpmu ∈ RealDomain(init = 2.94)
        kpsp ∈ RealDomain(init = 2.96)
        kpre ∈ RealDomain(init = 4)
        MW ∈ RealDomain(init = 349.317)
        logP ∈ RealDomain(init = 2.56)
        s_lumen ∈ RealDomain(init = 0.39*1000)
        L ∈ RealDomain(init = 280)
        d ∈ RealDomain(init = 2.5)
        PF ∈ RealDomain(init = 1.57)
        VF ∈ RealDomain(init = 6.5)
        MF ∈ RealDomain(init = 13)
```

```
        ITT ∈ RealDomain(init = 3.32)
        A ∈ RealDomain(init = 7440)
        B ∈ RealDomain(init = 1e7)
        alpha ∈ RealDomain(init = 0.6)
        beta ∈ RealDomain(init = 4.395)
        fabs ∈ RealDomain(init = 1)
        fdis ∈ RealDomain(init = 1)
        fperm ∈ RealDomain(init = 1)
        vad ∈ RealDomain(init = 18.2)
        vbo ∈ RealDomain(init =10.5)
        vbr ∈ RealDomain(init =1.45)
        vguWall ∈ RealDomain(init =0.65)
        vgulumen ∈ RealDomain(init =0.35)
        vhe ∈ RealDomain(init =0.33)
        vki ∈ RealDomain(init =0.31)
        vli ∈ RealDomain(init =1.8)
        vlu ∈ RealDomain(init =0.5)
        vmu ∈ RealDomain(init =29)
        vsp ∈ RealDomain(init =0.15)
        vbl ∈ RealDomain(init =5.6)
        FQad ∈ RealDomain(lower = 0.0, init = 0.05, upper = 1.0) #add bounds to
parameters for estimation
        FQbo ∈ RealDomain(lower = 0.0, init = 0.05, upper = 1.0)
        FQbr ∈ RealDomain(lower = 0.0, init = 0.12, upper = 1.0)
        FQgu ∈ RealDomain(lower = 0.0, init = 0.16, upper = 1.0)
        FQhe ∈ RealDomain(lower = 0.0, init = 0.04, upper = 1.0)
        FQki ∈ RealDomain(lower = 0.0, init = 0.19, upper = 1.0)
        FQli ∈ RealDomain(lower = 0.0, init = 0.255, upper = 1.0)
        FQmu ∈ RealDomain(lower = 0.0, init = 0.17, upper = 1.0)
        FQsp ∈ RealDomain(lower = 0.0, init = 0.03, upper = 1.0)
    end
    @pre begin
        Vgu = vguWall + vgulumen
        Vve = 0.705*vbl
        Var = 0.295*vbl
        Vre = WEIGHT - (vli+vki+vsp+vhe+vlu+vbo+vbr+vmu+vad+vguWall+vbl)
        CO = C_OUTPUT*60
        Qad = FQad*CO
        Qbo = FQbo*CO
        Qbr = FQbr*CO
        Qgu = FQgu*CO
        Qhe = FQhe*CO
        Qki = FQki*CO
        Qli = FQli*CO
        Qmu = FQmu*CO
        Qsp = FQsp*CO
        Qha = Qli - (Qgu+Qsp)
        Qtot = Qli+Qki+Qbo+Qhe+Qmu+Qad+Qbr
        Qre = CO - Qtot
        Qlu = CO
        Vgulumen = vgulumen
        S_lumen = s_lumen
        VguWall = vguWall
        Kpgu = kpgu
        BP = bp
        Vad = vad
        Kpad = kpad
        Vbr = vbr
        Kpbr = kpbr
```

```
        Vhe = vhe
        Kphe = kphe
        Vki = vki
        Kpki = kpki
        fup = Fup
        Vsp = vsp
        Kpsp = kpsp
        Vli = vli
        Kpli = kpli
        Vlu = vlu
        Kplu = kplu
        Kpmu = kpmu
        Kpre = kpre
        Vmu = vmu
        Vbl = vbl
        Vbo = vbo
        Kpbo = kpbo
        SA_abs = pi*L*d*PF*VF*MF*1e-4
        SA_basal = pi*L*d*PF*VF*1e-4
        MA = 10^logP
        MW_eff = MW - (3*17)
        Peff = fperm*A*(((MW_eff^(-alpha-beta))*MA)/((MW_eff^(-alpha)) +
B*(MW_eff^(-beta))*MA) * 1e-2 * 3600)
        kd = fdis*Peff*SA_abs*1000/vgulumen
        ka = fabs*Peff*SA_basal*1000/VguWall
        kt = 1/ITT
        scale_factor_H = MPPGL*Vli*1000
        scale_factor_G = MPPGI*VguWall*1000
        CLintHep = ((VmaxH/KmH)*scale_factor_H*60*1e-6)/fumic
        CLintGut = ((VmaxG/KmG)*scale_factor_G*60*1e-6)/fumic
        #CLintHep = CLintHep/fumic
        #CLintGut = CLintGut/fumic
        CLrenal = 0.096
        f = 1
    end
    @dynamics begin
        GUTLUMEN' = -kd*Vgulumen*(f*(GUTLUMEN/Vgulumen) + (1-f)*S_lumen) -
            kt*GUTLUMEN
        GUTWALL' = kd*Vgulumen*(f*(GUTLUMEN/Vgulumen) + (1-f)*S_lumen) -
            ka*GUTWALL - CLintGut*(GUTWALL/VguWall)
        GUT' = ka*GUTWALL + Qgu*((ART/Var) - (GUT/VguWall)/(Kpgu/BP))
        ADIPOSE' = Qad*((ART/Var) - (ADIPOSE/Vad)/(Kpad/BP))
        BRAIN' = Qbr*((ART/Var) - (BRAIN/Vbr)/(Kpbr/BP))
        HEART' = Qhe*((ART/Var) - (HEART/Vhe)/(Kphe/BP))
        KIDNEY' = Qki*((ART/Var) - (KIDNEY/Vki)/(Kpki/BP)) -
            CLrenal*(((KIDNEY/Vki)*fup)/(Kpki/BP))
        LIVER' = Qgu*((GUT/VguWall)/(Kpgu/BP)) + Qsp*((SPLEEN/Vsp)/(Kpsp/BP)) +
            Qha*(ART/Var) - Qli*((LIVER/Vli)/(Kpli/BP)) -
            CLintHep*(((LIVER/Vli)*fup)/(Kpli/BP))
        LUNG' = Qlu*((VEN/Vve) - (LUNG/Vlu)/(Kplu/BP))
        MUSCLE' = Qmu*((ART/Var) - (MUSCLE/Vmu)/(Kpmu/BP))
        SPLEEN' = Qsp*((ART/Var) - (SPLEEN/Vsp)/(Kpsp/BP))
        BONE' = Qbo*((ART/Var) - (BONE/Vbo)/(Kpbo/BP))
        REST' = Qre*((ART/Var) - (REST/Vre)/(Kpre/BP))
        VEN' = Qad*((ADIPOSE/Vad)/(Kpad/BP)) + Qbr*((BRAIN/Vbr)/(Kpbr/BP)) +
            Qhe*((HEART/Vhe)/(Kphe/BP)) + Qki*((KIDNEY/Vki)/(Kpki/BP)) +
            Qli*((LIVER/Vli)/(Kpli/BP)) + Qmu*((MUSCLE/Vmu)/(Kpmu/BP)) +
            Qbo*((BONE/Vbo)/(Kpbo/BP)) + Qre*((REST/Vre)/(Kpre/BP)) -
            Qlu*(VEN/Vve)
```

```
        ART' = Qlu*((LUNG/Vlu)/(Kplu/BP) - (ART/Var))
    end
    @derived begin
        Cvenn = VEN./Vve
        cp ~ @. Normal(Cvenn, 0.1) #for estimation
    end
end
```

```
PumasModel
  Parameters: Fup, fumic, WEIGHT, MPPGL, MPPGI, C_OUTPUT, VmaxH, VmaxG, KmH
, KmG, bp, kpad, kpbo, kpbr, kpgu, kphe, kpki, kpli, kplu, kpmu, kpsp, kpre
, MW, logP, s_lumen, L, d, PF, VF, MF, ITT, A, B, alpha, beta, fabs, fdis,
fperm, vad, vbo, vbr, vguWall, vgulumen, vhe, vki, vli, vlu, vmu, vsp, vbl,
 FQad, FQbo, FQbr, FQgu, FQhe, FQki, FQli, FQmu, FQsp
  Random effects:
  Covariates:
  Dynamical variables: GUTLUMEN, GUTWALL, GUT, ADIPOSE, BRAIN, HEART, KIDNE
Y, LIVER, LUNG, MUSCLE, SPLEEN, BONE, REST, VEN, ART
  Derived: Cvenn, cp
  Observed: Cvenn, cp
```

Let's create a subject to study the model

```
regimen_s = DosageRegimen(200, time=0, addl=13, ii=12, cmt=1, ss=1)
sub_s = Subject(id=1, events=regimen_s)
```

```
Subject
  ID: 1
  Events: 14
```

Below are setting the initial estimates of the parameters in the model

```
p = (Fup = 0.42, fumic = 0.711, WEIGHT = 73, MPPGL = 30.3, MPPGI = 0,
    C_OUTPUT = 6.5, VmaxH = 40, VmaxG = 40, KmH = 9.3, KmG = 9.3, bp = 1,
    kpad = 9.89, kpbo = 7.91, kpbr = 7.35, kpgu = 5.82, kphe = 1.95, kpki = 2.9,
    kpli = 4.66, kplu = 0.83, kpmu = 2.94, kpsp = 2.96, kpre = 4, MW = 349.317,
    logP = 2.56, s_lumen = 0.39*1000, L = 280, d = 2.5, PF = 1.57, VF = 6.5,
    MF = 13, ITT = 3.32, A = 7440, B = 1e7, alpha = 0.6, beta = 4.395, fabs = 1,
    fdis = 1, fperm = 1, vad = 18.2, vbo = 10.5, vbr = 1.45, vguWall = 0.65,
    vgulumen = 0.35, vhe = 0.33, vki = 0.31, vli = 1.8, vlu = 0.5, vmu = 29,
    vsp = 0.15, vbl = 5.6, FQad = 0.05, FQbo = 0.05, FQbr = 0.12, FQgu = 0.16,
    FQhe = 0.04, FQki = 0.19, FQli = 0.255, FQmu = 0.17, FQsp = 0.03)
```

```
(Fup = 0.42, fumic = 0.711, WEIGHT = 73, MPPGL = 30.3, MPPGI = 0, C_OUTPUT
= 6.5, VmaxH = 40, VmaxG = 40, KmH = 9.3, KmG = 9.3, bp = 1, kpad = 9.89, k
pbo = 7.91, kpbr = 7.35, kpgu = 5.82, kphe = 1.95, kpki = 2.9, kpli = 4.66,
 kplu = 0.83, kpmu = 2.94, kpsp = 2.96, kpre = 4, MW = 349.317, logP = 2.56
, s_lumen = 390.0, L = 280, d = 2.5, PF = 1.57, VF = 6.5, MF = 13, ITT = 3.
32, A = 7440, B = 1.0e7, alpha = 0.6, beta = 4.395, fabs = 1, fdis = 1, fpe
rm = 1, vad = 18.2, vbo = 10.5, vbr = 1.45, vguWall = 0.65, vgulumen = 0.35
, vhe = 0.33, vki = 0.31, vli = 1.8, vlu = 0.5, vmu = 29, vsp = 0.15, vbl =
 5.6, FQad = 0.05, FQbo = 0.05, FQbr = 0.12, FQgu = 0.16, FQhe = 0.04, FQki
 = 0.19, FQli = 0.255, FQmu = 0.17, FQsp = 0.03)
```

Let's take a look at the simulation of the model to ensure everything is working as expected.

```
simdata = simobs(model, [sub_s], p)
plot(simdata, obsnames=[:Cvenn])
```

```
Error: Cannot convert Pumas.SimulatedObservations{Pumas.Subject{NamedTuple{
(), Tuple{}}, Pumas.ConstantCovar{NamedTuple{(), Tuple{}}}, Vector{Pumas.Ev
```

```
ent{Float64, Float64, Float64, Float64, Float64, Float64, Int64}}, Nothing}
, StepRangeLen{Float64, Base.TwicePrecision{Float64}, Base.TwicePrecision{F
loat64}}, NamedTuple{(:Cvenn, :cp), Tuple{Vector{Float64}, Vector{Float64}}
}} to series data for plotting
```

We can run parameter estimation on the PBPK model with the `fit` function, we'll use the simulated data to run the estimation here `FQad`, `FQbo`, `FQbr`, `FQgu`, `FQhe`, `FQki`, `FQli`, `FQmu` and `FQsp` will be estimated within the bounds specified and the other parameters will be fixed.

```
data = read_pumas(DataFrame(simdata), observations = [:cp])
ft = fit(model, data, p, Pumas.NaivePooled(),
    constantcoef = (
        Fup = 0.42, fumic = 0.711, WEIGHT = 73, MPPGL = 30.3, MPPGI = 0,
        C_OUTPUT = 6.5, VmaxH = 40, VmaxG = 40, KmH = 9.3, KmG = 9.3, bp = 1,
        kpad = 9.89, kpbo = 7.91, kpbr = 7.35, kpgu = 5.82, kphe = 1.95,
        kpki = 2.9, kpli = 4.66, kplu = 0.83, kpmu = 2.94, kpsp = 2.96,
        kpre = 4, MW = 349.317, logP = 2.56, s_lumen = 0.39*1000, L = 280,
        d = 2.5, PF = 1.57, VF = 6.5, MF = 13, ITT = 3.32, A = 7440, B = 1e7,
        alpha = 0.6, beta = 4.395, fabs = 1, fdis = 1, fperm = 1, vad = 18.2,
        vbo = 10.5, vbr = 1.45, vguWall = 0.65, vgulumen = 0.35, vhe = 0.33,
        vki = 0.31, vli = 1.8, vlu = 0.5, vmu = 29, vsp = 0.15, vbl = 5.6),
    ensemblealg=EnsembleThreads())
```

```
Iter     Function value   Gradient norm
     0     3.532853e+03     2.748588e+02
 * time: 6.413459777832031e-5
     1     3.107116e+03     4.442407e+02
 * time: 0.4136669635772705
     2     2.227126e+03     5.066651e+02
 * time: 0.7907760143280029
     3     1.665027e+03     1.188483e+02
 * time: 1.17097806930542
     4     1.596237e+03     6.439351e+01
 * time: 1.5582449436187744
     5     1.477557e+03     1.059810e+02
 * time: 1.9403960704803467
     6     1.286222e+03     1.648540e+02
 * time: 2.407712936401367
     7     1.264300e+03     2.268807e+02
 * time: 2.990370035171509
     8     1.203290e+03     3.552203e+02
 * time: 3.4350991249084473
     9     1.020005e+03     1.616997e+02
 * time: 3.814635992050171
    10     9.661351e+02     1.106119e+02
 * time: 4.199761152267456
    11     9.502652e+02     3.324235e+01
 * time: 5.130814075469971
    12     9.375894e+02     1.405626e+01
 * time: 5.509473085403442
    13     9.331767e+02     8.551709e+00
 * time: 5.888031959533691
    14     9.316777e+02     8.287319e+00
 * time: 6.2693030834198
    15     9.310628e+02     9.826691e+00
 * time: 6.650858163833618
    16     9.304856e+02     1.202835e+01
 * time: 7.034079074859619
```

```
   17     9.296654e+02     1.438108e+01
* time: 7.418184995651245
   18     9.286640e+02     1.491707e+01
* time: 7.800615072250366
   19     9.280460e+02     1.207215e+01
* time: 8.182476997375488
   20     9.278836e+02     8.640973e+00
* time: 8.56185793876648
   21     9.277809e+02     5.309069e+00
* time: 8.94219708442688
   22     9.276883e+02     1.633804e+00
* time: 9.32240915298462
   23     9.276640e+02     8.767615e-01
* time: 9.704604148864746
   24     9.276611e+02     7.626913e-01
* time: 10.0868980884552
   25     9.276601e+02     4.552148e-01
* time: 10.468925952911377
   26     9.276590e+02     1.850559e-01
* time: 10.850022077560425
   27     9.276579e+02     4.139165e-01
* time: 11.231312036514282
   28     9.276553e+02     9.323739e-01
* time: 11.612580060958862
   29     9.276497e+02     1.577125e+00
* time: 11.994043111801147
   30     9.276369e+02     2.539629e+00
* time: 12.37494707107544
   31     9.276189e+02     3.571055e+00
* time: 12.755628108978271
   32     9.275978e+02     2.545767e+00
* time: 13.137118101119995
   33     9.275802e+02     9.523001e-01
* time: 13.517979145050049
   34     9.275795e+02     3.092575e-01
* time: 13.898752927780151
   35     9.275791e+02     4.972335e-01
* time: 14.280346155166626
   36     9.275787e+02     5.197397e-01
* time: 14.662178039550781
   37     9.275772e+02     4.408795e-01
* time: 15.043009996414185
   38     9.275746e+02     2.928362e-01
* time: 15.424016952514648
   39     9.275679e+02     5.753895e-01
* time: 15.805171966552734
   40     9.275531e+02     9.854275e-01
* time: 16.62190294265747
   41     9.275240e+02     2.138401e+00
* time: 16.99944806098938
   42     9.274954e+02     3.731443e+00
* time: 17.376451015472412
   43     9.274762e+02     2.720950e+00
* time: 17.751580953598022
   44     9.274630e+02     8.047854e-01
* time: 18.127344131469727
   45     9.274597e+02     7.693161e-01
* time: 18.529344081878662
   46     9.274590e+02     4.576667e-01
```

```
 * time: 18.93239712715149
    47     9.274588e+02     6.989313e-02
 * time: 19.335641145706177
    48     9.274587e+02     6.844361e-02
 * time: 19.721387147903442
    49     9.274578e+02     2.347732e-01
 * time: 20.09785008430481
    50     9.274567e+02     3.893690e-01
 * time: 20.478047132492065
    51     9.274540e+02     5.076756e-01
 * time: 20.860247135162354
    52     9.274502e+02     4.332910e-01
 * time: 21.242058038711548
    53     9.274482e+02     1.947036e-01
 * time: 21.623100996017456
    54     9.274478e+02     8.857160e-02
 * time: 22.0051691532135
    55     9.274474e+02     8.595779e-02
 * time: 22.388821125030518
    56     9.274466e+02     2.522982e-01
 * time: 22.8253071308136
    57     9.274448e+02     4.679212e-01
 * time: 23.20084309577942
    58     9.274406e+02     6.803278e-01
 * time: 23.576931953430176
    59     9.274321e+02     5.145327e-01
 * time: 23.955315113067627
    60     9.274237e+02     8.459128e-01
 * time: 24.332134008407593
    61     9.274224e+02     4.706659e-01
 * time: 24.705793142318726
    62     9.274221e+02     1.329927e-01
 * time: 25.081681966781616
    63     9.274221e+02     9.485953e-02
 * time: 25.45766806602478
    64     9.274220e+02     3.100899e-02
 * time: 25.83496403694153
    65     9.274220e+02     2.542107e-03
 * time: 26.209896087646484
    66     9.274220e+02     1.333987e-03
 * time: 26.586817026138306
    67     9.274220e+02     1.143827e-03
 * time: 26.967416048049927
    68     9.274220e+02     1.143827e-03
 * time: 28.06503701210022
FittedPumasModel

Successful minimization:                      true

Likelihood approximation:        Pumas.NaivePooled
Log-likelihood value:                   -927.42201
Number of subjects:                              1
Number of parameters:         Fixed      Optimized
                                 50              9
Observation records:         Active        Missing
    cp:                         181              0
    Total:                      181              0

-----------------------------------------------
```

	Estimate
Fup	0.42
fumic	0.711
WEIGHT	73.0
MPPGL	30.3
MPPGI	0.0
C_OUTPUT	6.5
VmaxH	40.0
VmaxG	40.0
KmH	9.3
KmG	9.3
bp	1.0
kpad	9.89
kpbo	7.91
kpbr	7.35
kpgu	5.82
kphe	1.95
kpki	2.9
kpli	4.66
kplu	0.83
kpmu	2.94
kpsp	2.96
kpre	4.0
MW	349.32
logP	2.56
s_lumen	390.0
L	280.0
d	2.5
PF	1.57
VF	6.5
MF	13.0
ITT	3.32
A	7440.0
B	1.0e7
alpha	0.6
beta	4.395
fabs	1.0
fdis	1.0
fperm	1.0
vad	18.2
vbo	10.5
vbr	1.45
vguWall	0.65
vgulumen	0.35
vhe	0.33
vki	0.31
vli	1.8
vlu	0.5
vmu	29.0
vsp	0.15
vbl	5.6
FQad	3.0466e-61
FQbo	2.1284999999999998e-50
FQbr	0.031432
FQgu	0.030545
FQhe	0.022844
FQki	0.028591
FQli	0.61246

```
FQmu                    0.25147
FQsp                    0.98398
-----------------------------------------------
```

0.1.2 GSA

We'll run the GSA on the AUC and Cmax output of the `Cvenn` variable and therefore redefine the model to include the NCA calculation.

```
model = @model begin
    @param begin
        Fup ∈ RealDomain(init = 0.42)
        fumic ∈ RealDomain(init = 0.711)
        WEIGHT ∈ RealDomain(init = 73)
        MPPGL ∈ RealDomain(init = 30.3)
        MPPGI ∈ RealDomain(init = 0)
        C_OUTPUT ∈ RealDomain(init = 6.5)
        VmaxH ∈ RealDomain(init = 40)
        VmaxG ∈ RealDomain(init = 40)
        KmH ∈ RealDomain(init = 9.3)
        KmG ∈ RealDomain(init = 9.3)
        bp ∈ RealDomain(init = 1)
        kpad ∈ RealDomain(init = 9.89)
        kpbo ∈ RealDomain(init = 7.91)
        kpbr ∈ RealDomain(init = 7.35)
        kpgu ∈ RealDomain(init = 5.82)
        kphe ∈ RealDomain(init = 1.95)
        kpki ∈ RealDomain(init = 2.9)
        kpli ∈ RealDomain(init = 4.66)
        kplu ∈ RealDomain(init = 0.83)
        kpmu ∈ RealDomain(init = 2.94)
        kpsp ∈ RealDomain(init = 2.96)
        kpre ∈ RealDomain(init = 4)
        MW ∈ RealDomain(init = 349.317)
        logP ∈ RealDomain(init = 2.56)
        s_lumen ∈ RealDomain(init = 0.39*1000)
        L ∈ RealDomain(init = 280)
        d ∈ RealDomain(init = 2.5)
        PF ∈ RealDomain(init = 1.57)
        VF ∈ RealDomain(init = 6.5)
        MF ∈ RealDomain(init = 13)
        ITT ∈ RealDomain(init = 3.32)
        A ∈ RealDomain(init = 7440)
        B ∈ RealDomain(init = 1e7)
        alpha ∈ RealDomain(init = 0.6)
        beta ∈ RealDomain(init = 4.395)
        fabs ∈ RealDomain(init = 1)
        fdis ∈ RealDomain(init = 1)
        fperm ∈ RealDomain(init = 1)
        vad ∈ RealDomain(init = 18.2)
        vbo ∈ RealDomain(init =10.5)
        vbr ∈ RealDomain(init =1.45)
        vguWall ∈ RealDomain(init =0.65)
        vgulumen ∈ RealDomain(init =0.35)
        vhe ∈ RealDomain(init =0.33)
        vki ∈ RealDomain(init =0.31)
        vli ∈ RealDomain(init =1.8)
        vlu ∈ RealDomain(init =0.5)
        vmu ∈ RealDomain(init =29)
```

```
    vsp ∈ RealDomain(init =0.15)
    vbl ∈ RealDomain(init =5.6)
    FQad ∈ RealDomain(lower = 0.0, init = 0.05, upper = 1.0)
    FQbo ∈ RealDomain(lower = 0.0, init = 0.05, upper = 1.0)
    FQbr ∈ RealDomain(lower = 0.0, init = 0.12, upper = 1.0)
    FQgu ∈ RealDomain(lower = 0.0, init = 0.16, upper = 1.0)
    FQhe ∈ RealDomain(lower = 0.0, init = 0.04, upper = 1.0)
    FQki ∈ RealDomain(lower = 0.0, init = 0.19, upper = 1.0)
    FQli ∈ RealDomain(lower = 0.0, init = 0.255, upper = 1.0)
    FQmu ∈ RealDomain(lower = 0.0, init = 0.17, upper = 1.0)
    FQsp ∈ RealDomain(lower = 0.0, init = 0.03, upper = 1.0)
end
@pre begin
    Vgu = vguWall + vgulumen
    Vve = 0.705*vbl
    Var = 0.295*vbl
    Vre = WEIGHT - (vli+vki+vsp+vhe+vlu+vbo+vbr+vmu+vad+vguWall+vbl)
    CO = C_OUTPUT*60
    Qad = FQad*CO
    Qbo = FQbo*CO
    Qbr = FQbr*CO
    Qgu = FQgu*CO
    Qhe = FQhe*CO
    Qki = FQki*CO
    Qli = FQli*CO
    Qmu = FQmu*CO
    Qsp = FQsp*CO
    Qha = Qli - (Qgu+Qsp)
    Qtot = Qli+Qki+Qbo+Qhe+Qmu+Qad+Qbr
    Qre = CO - Qtot
    Qlu = CO
    Vgulumen = vgulumen
    S_lumen = s_lumen
    VguWall = vguWall
    Kpgu = kpgu
    BP = bp
    Vad = vad
    Kpad = kpad
    Vbr = vbr
    Kpbr = kpbr
    Vhe = vhe
    Kphe = kphe
    Vki = vki
    Kpki = kpki
    fup = Fup
    Vsp = vsp
    Kpsp = kpsp
    Vli = vli
    Kpli = kpli
    Vlu = vlu
    Kplu = kplu
    Kpmu = kpmu
    Kpre = kpre
    Vmu = vmu
    Vbl = vbl
    Vbo = vbo
    Kpbo = kpbo
    SA_abs = pi*L*d*PF*VF*MF*1e-4
    SA_basal = pi*L*d*PF*VF*1e-4
```

```
        MA = 10^logP
        MW_eff = MW - (3*17)
        Peff = fperm*A*(((MW_eff^(-alpha-beta))*MA)/((MW_eff^(-alpha)) +
B*(MW_eff^(-beta))*MA) * 1e-2 * 3600)
        kd = fdis*Peff*SA_abs*1000/vgulumen
        ka = fabs*Peff*SA_basal*1000/VguWall
        kt = 1/ITT
        scale_factor_H = MPPGL*Vli*1000
        scale_factor_G = MPPGI*VguWall*1000
        CLintHep = ((VmaxH/KmH)*scale_factor_H*60*1e-6)/fumic
        CLintGut = ((VmaxG/KmG)*scale_factor_G*60*1e-6)/fumic
        #CLintHep = CLintHep/fumic
        #CLintGut = CLintGut/fumic
        CLrenal = 0.096
        f = 1
    end
    @dynamics begin
        GUTLUMEN' = -kd*Vgulumen*(f*(GUTLUMEN/Vgulumen) + (1-f)*S_lumen) -
            kt*GUTLUMEN
        GUTWALL' = kd*Vgulumen*(f*(GUTLUMEN/Vgulumen) + (1-f)*S_lumen) -
            ka*GUTWALL - CLintGut*(GUTWALL/VguWall)
        GUT' = ka*GUTWALL + Qgu*((ART/Var) - (GUT/VguWall)/(Kpgu/BP))
        ADIPOSE' = Qad*((ART/Var) - (ADIPOSE/Vad)/(Kpad/BP))
        BRAIN' = Qbr*((ART/Var) - (BRAIN/Vbr)/(Kpbr/BP))
        HEART' = Qhe*((ART/Var) - (HEART/Vhe)/(Kphe/BP))
        KIDNEY' = Qki*((ART/Var) - (KIDNEY/Vki)/(Kpki/BP)) -
            CLrenal*(((KIDNEY/Vki)*fup)/(Kpki/BP))
        LIVER' = Qgu*((GUT/VguWall)/(Kpgu/BP)) + Qsp*((SPLEEN/Vsp)/(Kpsp/BP)) +
            Qha*(ART/Var) - Qli*((LIVER/Vli)/(Kpli/BP)) -
            CLintHep*(((LIVER/Vli)*fup)/(Kpli/BP))
        LUNG' = Qlu*((VEN/Vve) - (LUNG/Vlu)/(Kplu/BP))
        MUSCLE' = Qmu*((ART/Var) - (MUSCLE/Vmu)/(Kpmu/BP))
        SPLEEN' = Qsp*((ART/Var) - (SPLEEN/Vsp)/(Kpsp/BP))
        BONE' = Qbo*((ART/Var) - (BONE/Vbo)/(Kpbo/BP))
        REST' = Qre*((ART/Var) - (REST/Vre)/(Kpre/BP))
        VEN' = Qad*((ADIPOSE/Vad)/(Kpad/BP)) + Qbr*((BRAIN/Vbr)/(Kpbr/BP)) +
            Qhe*((HEART/Vhe)/(Kphe/BP)) + Qki*((KIDNEY/Vki)/(Kpki/BP)) +
            Qli*((LIVER/Vli)/(Kpli/BP)) + Qmu*((MUSCLE/Vmu)/(Kpmu/BP)) +
            Qbo*((BONE/Vbo)/(Kpbo/BP)) + Qre*((REST/Vre)/(Kpre/BP)) -
            Qlu*(VEN/Vve)
        ART' = Qlu*((LUNG/Vlu)/(Kplu/BP) - (ART/Var))
    end
    @derived begin
        Cvenn = VEN./Vve
        #capturing NCA metrics for evaluations
        nca := @nca Cvenn
        auc =  last(NCA.auc(nca))
        cmax = last(NCA.cmax(nca))
    end
end

PumasModel
  Parameters: Fup, fumic, WEIGHT, MPPGL, MPPGI, C_OUTPUT, VmaxH, VmaxG, KmH
, KmG, bp, kpad, kpbo, kpbr, kpgu, kphe, kpki, kpli, kplu, kpmu, kpsp, kpre
, MW, logP, s_lumen, L, d, PF, VF, MF, ITT, A, B, alpha, beta, fabs, fdis,
fperm, vad, vbo, vbr, vguWall, vgulumen, vhe, vki, vli, vlu, vmu, vsp, vbl,
 FQad, FQbo, FQbr, FQgu, FQhe, FQki, FQli, FQmu, FQsp
  Random effects:
  Covariates:
  Dynamical variables: GUTLUMEN, GUTWALL, GUT, ADIPOSE, BRAIN, HEART, KIDNE
```

```
Y, LIVER, LUNG, MUSCLE, SPLEEN, BONE, REST, VEN, ART
  Derived: Cvenn, auc, cmax
  Observed: Cvenn, auc, cmax
```

To run the GSA we'll define the parameter ranges for our parameters of interest.

```
p_range_low = (fperm=1/3, s_lumen=390/3, ITT = 3.32/3, MPPGI=1.44/3, )

p_range_high = (fperm=1*3, s_lumen=390*3, ITT = 3.32*3, MPPGI=1.44*3, )
```

```
(fperm = 3, s_lumen = 1170, ITT = 9.959999999999999, MPPGI = 4.32)
```

Now, we are ready to run GSA on our model.

The Sobol Method We will run the Sobol method for 1000 iterations, please note that this takes a couple of hours to finish because of the complexity of the model.

```
sobol_ = Pumas.gsa(model, sub_s, p, Pumas.Sobol(), [:cmax,:auc],
p_range_low,p_range_high, N=1000, obstimes=0:1:300)
```

```
Error: type Nothing has no field time
```

We can use scatter plot the result to visualize the result.

```
keys_ = keys(p_range_low)
cmax_s1 = [sobol_.first_order[1,:][key] for key in keys_]
cmax_st = [sobol_.total_order[1,:][key] for key in keys_]

plot_cmax = scatter([string.(keys_)...], cmax_s1, ylims = (0,1), label = "First
Order",title="Cmax")
scatter!(plot_cmax,[string.(keys_)...], cmax_st, ylims = (0,1), label = "Total Order",
marker=:utriangle)

auc_s1 = [sobol_.first_order[2,:][key] for key in keys_]
auc_st = [sobol_.total_order[2,:][key] for key in keys_]
plot_auc = scatter([string.(keys_)...], auc_s1, ylims = (0,1), label = "First Order",
title="AUC")
scatter!(plot_auc, [string.(keys_)...], auc_st, ylims = (0,1), label = "Total Order",
marker=:utriangle)
plot(plot_cmax, plot_auc, size = (1200,400))
```

```
Error: UndefVarError: sobol_ not defined
```

0.1.3 The eFAST method

eFAST method allows the estimation of first order and total Sobol indices in a more computationaly efficient way.

```
eFAST_ = Pumas.gsa(model, sub_s, p, Pumas.eFAST(), [:cmax,:auc], p_range_low,
p_range_high, n=1000, obstimes=0:1:300)
```

```
Error: type Nothing has no field time
```

We can use scatter plot the result to visualize the result.

```
keys_ = keys(p_range_low)
cmax_s1 = [eFAST_.first_order[1,:][key] for key in keys_]
cmax_st = [eFAST_.total_order[1,:][key] for key in keys_]
```

```
plot_cmax = scatter([string.(keys_)...], cmax_s1, ylims = (0,1), label = "First
Order",title="Cmax")
scatter!(plot_cmax,[string.(keys_)...], cmax_st, ylims = (0,1), label = "Total Order",
marker=:utriangle)

auc_s1 = [eFAST_.first_order[2,:][key] for key in keys_]
auc_st = [eFAST_.total_order[2,:][key] for key in keys_]
plot_auc = scatter([string.(keys_)...], auc_s1, ylims = (0,1), label = "First Order",
title="AUC")
scatter!(plot_auc,[string.(keys_)...], auc_st, ylims = (0,1), label = "Total Order",
marker=:utriangle)
plot(plot_cmax, plot_auc, size = (1200,400))
```

```
Error: UndefVarError: eFAST_ not defined
```

0.2 Conclusion

We observe for both AUC and Cmax `fperm` and `MPPGI` show high values for both First and Total Order indices of Sobol whereas `s_lumen` and `ITT` have no effect at all and show a value of zero for the indices.