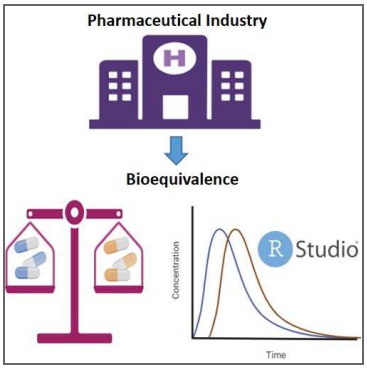

JAN 08, 2024

OPEN ACCESS

DOI:
dx.doi.org/10.17504/protocols.io.q26g7p7d1gwz/v1

Protocol Citation: Leandro do Prado Assunção, Laura Raniere Borges dos Anjos 2024. Protocol for bioequivalence analysis in studies with two treatments (design 2x2x2, 2x3x3 and 2x2x4) using the R Studio software. **protocols.io** https://dx.doi.org/10.17504/protocols.io.q26g7p7d1gwz/v1

Protocol status: Working
We use this protocol and it's working

Protocol for bioequivalence analysis in studies with two treatments (design 2x2x2, 2x3x3 and 2x2x4) using the R Studio software

Leandro do Prado Assunção[1], Laura Raniere Borges dos Anjos[1]

[1]Instituto de Ensino Estatístico e Científico (IEEC)

IEEC

Leandro do Prado Assunção

Created: Jan 07, 2024

Last Modified: Jan 08, 2024

PROTOCOL integer ID: 93031

Keywords: Bioequivalence, Pharmacokinetic Parameters, Cmáx, AUC, Design Study 2x2x2, Design Study 2x3x3, Design Study 2x2x4, R Studio

ABSTRACT

Research and development (R&D) of new drugs is a complex process that requires high financial investments and involves risk of failure and a long time. When the pharmaceutical industry discovers a new active compounds, it is inherent that this innovative drug is produced and marketed, for a few years, only by said laboratory. Only after the end of the patent period is that the formula of such a drug is shared with other laboratories that have an interest in creating generic formulas of this active compounds. However, before the production of the generic it is necessary to verify the biological equivalence (*in vitro* e *in vivo*) of the generic drug (test) with the reference drug. This comparative study, which aims to verify whether a drug test is the therapeutic equivalent of the reference drug, is called bioequivalence. A drug test is considered generic or similar when it presents a pharmacokinetic profile similar to the reference drug. This process appears to be simple, but it is not and there are equivalence standards that need to be followed for the drug to be approved as bioequivalent. These standards are defined by regulatory agencies distributed around the world. In the United States, for example, the regulatory agency is the Food and Drug Administration (FDA). In Europe, the regulatory agency is the European Medicines Agency (EMA). In Brazil, the Agencia Nacional de Vigilância Sanitária (ANVISA) is responsible for ensuring the bioequivalence of drugs, in addition to ensuring that they are produced using good manufacturing practices. These bioequivalence studies involve a multidisciplinary team that includes doctors, nurses, biomedical, biologists, biotechnologists, chemicals, pharmaceuticals, statistics, among others. The statistical analysis team, in particular, is responsible for the planning and design of the study, as well as the performance of data analysis and pointing out its interpretations. These statistical analyses are performed from software that is not free and requires license renewal. On the other hand, RStudio is free software and has been widely used for various types of statistical analysis. Here, we have created a practical and basic guide to help you calculate, using the RStudio software, the parameters Cmáx, AUC e Tmáx that are necessary for bioequivalence analysis. In addition, we provide the algorithms for the development of bioequivalence analysis for two treatment with the calculation of geometric means, ratio, confidence interval 90%, intra-subject CV, inter-subject CV, power TOST and analysis of variability (CVwr e CVwt). It is important to note that the algorithms presented here were developed from libraries already available for free in the RStudio software and their references were informed. We believe that the availability of these algorithms can contribute to further bioequivalence studies and thus stimulate the generation of knowledge in relation to pharmacokinetic parameters.

Installing the packages

1 The default function in R Studio to install packages is install.packages("Name"). Run the code.

1.1 Command:

```
install.packages("readxl")
install.packages("dplyr")
install.packages("PKNCA")
install.packages("stringr")
install.packages("ggplot2")
install.packages("openxlsx")
install.packages("nlme")
install.packages("dvmisc")
install.packages("PowerTOST")
install.packages("BE")
install.packages("replicateBE")
```

Loading the Packages

2 The default function in R Studio to loading packages is library(Name). Run the code.

2.1 Command:

```
library(readxl)
library(dplyr)
library(PKNCA)
library(stringr)
library(ggplot2)
library(openxlsx)
library(nlme)
library(dvmisc)
library(PowerTOST)
library(BE)
library(replicateBE)
```

Function for Calculation of pharmacokinetic parameters (PK)

3 Run the code:

```
calc_PK = function(BD, dose){
 colnames(BD) = c("Subject",
                  "Formulation",
                  "Time",
                  "Concentration",
                  "Sequence",
                  "Period") # Renaming the variables of the Data.frame

 BD$Time = as.numeric(BD$Time) # Transforming the variable into number type
 BD$Concentration = as.numeric(BD$Concentration)  # Transforming the variable into number type
 BD$Dose = dose  # Dose used in the study

 dat_PK_final = data.frame(subj = c(), auc = c(), cmax = c(), tmax = c(),
                           per = c(), Seq = c(), trat = c()) # Creating a new data.frame

 qtq_per = as.vector(unique(BD$Period)) # Period vector

 # Loop to calculate pharmacokinetic parameters by period #

 for(i in 1:length(qtq_per)){

  period = qtq_per[i]

  ### Sequencing and treatment by period ###

  dat1 = filter(BD, Period == period) # Filter by period

  sujeitos = data.frame(table(dat1$Subject))
  individuos = as.vector(sujeitos$Var1)

  ind = c()
  seq = c()
  trat = c()

  for(i in 1:length(individuos)){
   dat_st = filter(dat1, Subject == individuos[i])
   ind[i] = individuos[i]
   seq[i] = dat_st[1,5]
   trat[i] = dat_st[1,2]
  }

  seq_trat1 = data.frame(cbind(ind, seq, trat))

  ### Calculations of pharmacokinetic parameters in the period ###
```

```
  concetration = PKNCAconc(as.data.frame(dat1), Concentration~Time|Subject)
  dose_adm1 = unique(dat1[dat1$Time == 0, c("Dose", "Time", "Subject")])
  dose_adm2 = PKNCAdose(dose_adm1, Dose~Time|Subject)
  data_1 = PKNCAdata(concetration, dose_adm2)
  results_PK = pk.nca(data_1)
  parameters_PK = data.frame(results_PK$result)
  subject = data.frame(table(parameters_PK$Subject))
  subject1 = as.vector(subject$Var1)
  table_PK = select(parameters_PK, Subject, PPTESTCD, PPORRES)

  subj = c()
  auc = c()
  cmax = c()
  tmax = c()
  per = c()

  for(i in 1:length(subject1)){
    dat = filter(table_PK, Subject == subject1[i])
    subj[i] = subject1[i]
    auc[i] = dat[1,3]
    cmax[i] = dat[2,3]
    tmax[i] = dat[3,3]
    per[i] = period
  }

  dat_PK = data.frame(subj, auc = round(auc,3), cmax = round(cmax,3),
                      tmax = round(tmax,3), per, Seq = seq_trat1$seq,
                      trat = seq_trat1$trat)

  dat_PK_final = rbind(dat_PK_final, dat_PK)
}

### Working with the subjects ###

n = data.frame(strsplit(dat_PK_final$subj, ". "))
part = c()
for(i in 1:length(n)){
  part[i] = n[2,i]
}
part = as.numeric(part)
dat_PK_final$subj = part

dat_PK_final1 = select(dat_PK_final, SUBJ = subj,
```

```
                    GRP = Seq, PRD = per, TRT = trat,
                    AUClast = auc, Cmax = cmax, Tmax = tmax)

  # Standardization of sequence and formulation in Test (T) and Reference (R) #

  dat_PK_final1[dat_PK_final1 == "AB"] = "RT"
  dat_PK_final1[dat_PK_final1 == "BA"] = "TR"
  dat_PK_final1[dat_PK_final1 == "A"] = "R"
  dat_PK_final1[dat_PK_final1 == "B"] = "T"
  dat_PK_final1$GRP = as.factor(dat_PK_final1$GRP)
  dat_PK_final1$TRT = as.factor(dat_PK_final1$TRT)

  ### Treatment and Active Graphic Mean ###

  data1 = aggregate(Concentration ~ Formulation + Time, BD, mean)

  g1 = ggplot(data = data1 ,aes(x = Time, y = Concentration)) +
   geom_line(aes(colour = Formulation, group = Formulation)) +
   geom_point() +
   labs(title = "Average graph of pharmacokinetic profile treatment and test",
        x = "Time (hours)", y = "Mean concentration of the analyte (ug/mL)") +
   theme_bw() + theme_classic() + theme(text = element_text(size = 12))

  print(g1)
  return(dat_PK_final1)
 }
```

Example of application with hypothetical data and dose equa...

4 Command:

```
dat = openxlsx::read.xlsx("dados_hipoteticos.xlsx", sheet = 1) # Loading of date
calc_PK(dat, 10) # Application of function calc_PK
```

Function to Calculate Ratio, Confidence Interval (IC) 90%, CVi...

5 Command:

```
bioequivalencia_f2_V1 = function(dados, desenho){
 if(desenho == "2x2"){
  PK = dados
  PK$TRT = as.factor(PK$TRT) # Transforming the variable treatment into a factor
  PK$TRT = relevel(PK$TRT, ref="R") # Select treatment reference as comparator
  tratR = PK %>% filter(TRT == "R") # Filter of the values Reference
```

```
tratT = PK %>% filter(TRT == "T") # Filter of the values Test
n1 = length(tratR$SUBJ) # Sample size of the Reference
n2 = length(tratT$SUBJ) # Sample size of the Test

# Calculation of the geometrical means #

med_geom = function(vetor){
  media_g = round(prod(vetor)^(1/length(vetor)),5)
  return(media_g)
}

# Calculation of the geometrical mean of Cmáx and AUC #

med_R_auc = med_geom(tratR[,5])
med_T_auc = med_geom(tratT[,5])

med_R_cmax = med_geom(tratR[,6])
med_T_cmax = med_geom(tratT[,6])

# AUC model for to meet the IC90%, CVintra and power TOST #

Modelo_AUC_1 = lme(log(AUClast) ~ TRT + GRP + PRD, random=~1|SUBJ, data=PK) # Use of the lme function
varia_AUC = data.frame(VarCorr(Modelo_AUC_1)) # Variability Matrix
CVinter_AUC = (sqrt(exp(as.numeric(varia_AUC[1,1][1]))-1))*100 # Calculate of CVinter
CVintra_AUC = (sqrt(exp(as.numeric(varia_AUC[2,1][1]))-1))*100 # Calculate of CVintra
Type_III_AUC = anova(Modelo_AUC_1) # Calculate of efects
attr(Type_III_AUC, "heading")[1] = "Type III Analysis of Variance Table\n"
ci_AUC = intervals(Modelo_AUC_1, 0.90, which="fixed") # Confidence Interval (CI) 90%

RxT_AUC = data.frame(Estimativa = round(exp(ci_AUC$fixed["TRTT",]),5)) # Calculate of Ratio and CI

Ratio_AUC = as.numeric(RxT_AUC[2,1]) # Ratio of AUC

Poder_AUC = 100*power.TOST(alpha = 0.05, logscale = TRUE,
                    theta0 = Ratio_AUC, theta1 = 0.80,
                    theta2=1.25, n=c(n1, n2),
                    CV=(CVintra_AUC/100),
                    design = "2x2") # Calculate of Power TOST for AUC

# Model of Cmáx for to meet  CI90%, CVintra and power TOST

Modelo_Cmax_1 = lme(log(Cmax) ~ TRT + GRP + PRD, random=~1|SUBJ, data=PK) # Use of function lme
```

```
  varia_Cmax = data.frame(VarCorr(Modelo_Cmax_1)) # Matrix of Variability
  CVinter_Cmax = (sqrt(exp(as.numeric(varia_Cmax[1,1][1]))-1))*100 # Calculate of CVinter
  CVintra_Cmax = (sqrt(exp(as.numeric(varia_Cmax[2,1][1]))-1))*100 # Calculate of  CVintra
  Type_III_Cmax = anova(Modelo_Cmax_1) # Calculate of efects
  attr(Type_III_Cmax, "heading")[1] = "Type III Analysis of Variance Table\n"
  ci_Cmax = intervals(Modelo_Cmax_1, 0.90, which="fixed") # CI 90%

  RxT_Cmax = data.frame(Estimativa = round(exp(ci_Cmax$fixed["TRTT",]),5)) # Calculate Ratio and CI
90%

  Ratio_Cmax = as.numeric(RxT_Cmax[2,1]) # Ratio of Cmáx

  Poder_Cmax = 100*power.TOST(alpha = 0.05, logscale = TRUE,
                      theta0 = Ratio_Cmax, theta1 = 0.80,
                      theta2=1.25, n=c(n1,n2), CV=(CVintra_Cmax/100),
                      design = "2x2") # Calculate of  Power TOST for Cmax

  # Table of Results with sample size, Ratio, CI 90%, CVinter, CVintra and Power TOST

  result = data.frame(PK = c("Cmax", "AUC"),
                n_A = c(n1, n2),
                n_B = c(n1, n2),
                Ratio = c(Ratio_Cmax*100, Ratio_AUC*100),
                LL = c(RxT_Cmax[1,1]*100, RxT_AUC[1,1]*100),
                UL = c(RxT_Cmax[3,1]*100, RxT_AUC[3,1]*100),
                CV_Between = c(CVinter_Cmax, CVinter_AUC),
                CV_Within = c(CVintra_Cmax, CVintra_AUC),
                Power_TOST = c(Poder_Cmax, Poder_AUC))

  return(result)

 }else if(desenho == "2x3"){ # For design of study 2x3x3
  PK = dados
  PK$TRT = as.factor(PK$TRT) # Transforming the variable treatment into a factor
  PK$TRT = relevel(PK$TRT, ref="R") # Select reference treatment as comparator
  tratR = PK %>% filter(TRT == "R") # Filter of the values Reference
  tratT = PK %>% filter(TRT == "T") # Filter of the values Test
  n1 = length(tratR$SUBJ) # Simple Size of Reference
  n2 = length(tratT$SUBJ) # Simple Size of Test

  # Calculate of geometrical means

  med_geom = function(vetor){
   media_g = round(prod(vetor)^(1/length(vetor)),5)
```

```
  return(media_g)
}

# Calculate of geometrical means of Cmáx and AUC #

med_R_auc = med_geom(tratR[,5])
med_T_auc = med_geom(tratT[,5])

med_R_cmax = med_geom(tratR[,6])
med_T_cmax = med_geom(tratT[,6])

# Model of AUC for to meet CI 90%, CVintra and power TOST

Modelo_AUC_1 = lme(log(AUClast) ~ TRT + GRP + PRD, random=~1|SUBJ, data=PK) # Use of function
lme
varia_AUC = data.frame(VarCorr(Modelo_AUC_1)) # Matrix of Variability
CVinter_AUC = (sqrt(exp(as.numeric(varia_AUC[1,1][1]))-1))*100 # Calculate of CVinter
CVintra_AUC = (sqrt(exp(as.numeric(varia_AUC[2,1][1]))-1))*100 # Calculate of CVintra
Type_III_AUC = anova(Modelo_AUC_1) # Calculate of efects
attr(Type_III_AUC, "heading")[1] = "Type III Analysis of Variance Table\n"
ci_AUC = intervals(Modelo_AUC_1, 0.90, which="fixed") # CI 90%

RxT_AUC = data.frame(Estimativa = round(exp(ci_AUC$fixed["TRTT",]),5)) # Calculate Ratio and CI 90%

Ratio_AUC = as.numeric(RxT_AUC[2,1]) # Ratio of AUC

Poder_AUC = 100*power.TOST(alpha = 0.05, logscale = TRUE,
                    theta0 = Ratio_AUC, theta1 = 0.80,
                    theta2=1.25, n=c(n1, n2),
                    CV=(CVintra_AUC/100),
                    design = "2x2") # Calculate of Power TOST for AUC

# Model of Cmáx for to meet CI 90%, CVintra and Power TOST

Modelo_Cmax_1 = lme(log(Cmax) ~ TRT + GRP + PRD, random=~1|SUBJ, data=PK) # Use of function
lme
varia_Cmax = data.frame(VarCorr(Modelo_Cmax_1)) # Matrix of Variability
CVinter_Cmax = (sqrt(exp(as.numeric(varia_Cmax[1,1][1]))-1))*100 # Calculate of CVinter
CVintra_Cmax = (sqrt(exp(as.numeric(varia_Cmax[2,1][1]))-1))*100 # Calculate of CVintra
Type_III_Cmax = anova(Modelo_Cmax_1) # Calculate of efects
attr(Type_III_Cmax, "heading")[1] = "Type III Analysis of Variance Table\n"
ci_Cmax = intervals(Modelo_Cmax_1, 0.90, which="fixed") # CI 90%

RxT_Cmax = data.frame(Estimativa = round(exp(ci_Cmax$fixed["TRTT",]),5)) # Calculate Ratio and CI
```

```
90%

   Ratio_Cmax = as.numeric(RxT_Cmax[2,1]) # Ratio of Cmáx

   Poder_Cmax = 100*power.TOST(alpha = 0.05, logscale = TRUE,
                        theta0 = Ratio_Cmax, theta1 = 0.80,
                        theta2=1.25, n=c(n1,n2), CV=(CVintra_Cmax/100),
                        design = "2x2") # Calculate of Power TOST for Cmax

   # Table of Results With Simple Size, Ratio, CI 90%, CVinter, CVintra and Power TOST

   result = data.frame(PK = c("Cmax", "AUC"),
                       n_A = c(n1, n2),
                       n_B = c(n1, n2),
                       Ratio = c(Ratio_Cmax*100, Ratio_AUC*100),
                       LL = c(RxT_Cmax[1,1]*100, RxT_AUC[1,1]*100),
                       UL = c(RxT_Cmax[3,1]*100, RxT_AUC[3,1]*100),
                       CV_Between = c(CVinter_Cmax, CVinter_AUC),
                       CV_Within = c(CVintra_Cmax, CVintra_AUC),
                       Power_TOST = c(Poder_Cmax, Poder_AUC))

   # Analysis of Variability for to calculate CVwr and CVwt #

   # For ASC0-t #

   dat_Ref_ASC = PK %>% filter(TRT == "R") # Separating the data Reference
   model_CVwr_ASC  <- lm(log(AUClast) ~ GRP + SUBJ %in% SUBJ + PRD, dat_Ref_ASC) # Models with
ASC Reference data only
   CVwr_ASC  = round(sqrt(exp(anova(model_CVwr_ASC)["Residuals", "Mean Sq"]) - 1)*100, 3) # Estimate
of CVwr of ASC - Reference

   # For Cmax #

   dat_Ref_Cmax = PK %>% filter(TRT == "R") # Separating the data Reference
   model_CVwr_Cmax <- lm(log(Cmax) ~ GRP + SUBJ %in% SUBJ + PRD, dat_Ref_Cmax) # Models with
Cmax Reference data only
   CVwr_Cmax = round(sqrt(exp(anova(model_CVwr_Cmax)["Residuals", "Mean Sq"]) - 1)*100, 3) #
Estimate of CVwr of Cmax - Reference

   result1 = result %>% mutate(CVwr = c(CVwr_Cmax, CVwr_ASC)) # Results of the variability analysis

   return(result1)

  }else if(desenho == "2x4"){
```

```
PK = dados
PK$TRT = as.factor(PK$TRT) # Transforming the variable treatment into a factor
PK$TRT = relevel(PK$TRT, ref="R") # Select referential treatment as comparator
tratR = PK %>% filter(TRT == "R") # Filter of the values Reference
tratT = PK %>% filter(TRT == "T") #  Filter of the values Test
n1 = length(tratR$SUBJ) # Simple Size of Reference
n2 = length(tratT$SUBJ) # Simple Size of Test

# Calculate  of the Mean geometricas

med_geom = function(vetor){
  media_g = round(prod(vetor)^(1/length(vetor)),5)
  return(media_g)
}

# Calculate  of the Mean geometricas of Cmáx and AUC #

med_R_auc = med_geom(tratR[,5])
med_T_auc = med_geom(tratT[,5])

med_R_cmax = med_geom(tratR[,6])
med_T_cmax = med_geom(tratT[,6])

# Model of AUC for meet  CI 90%, CVintra and power TOST

Modelo_AUC_1 = lme(log(AUClast) ~ TRT + GRP + PRD, random=~1|SUBJ, data=PK) # Use of function
lme
varia_AUC = data.frame(VarCorr(Modelo_AUC_1)) # Matrix of Variability
CVinter_AUC = (sqrt(exp(as.numeric(varia_AUC[1,1][1]))-1))*100 # Calculate of CVinter
CVintra_AUC = (sqrt(exp(as.numeric(varia_AUC[2,1][1]))-1))*100 # Calculate of CVintra
Type_III_AUC = anova(Modelo_AUC_1) # Calculate of efects
attr(Type_III_AUC, "heading")[1] = "Type III Analysis of Variance Table\n"
ci_AUC = intervals(Modelo_AUC_1, 0.90, which="fixed") # CI 90%

RxT_AUC = data.frame(Estimativa = round(exp(ci_AUC$fixed["TRTT",]),5)) # Calculate Ratio and CI 90%

Ratio_AUC = as.numeric(RxT_AUC[2,1]) # Ratio of AUC

Poder_AUC = 100*power.TOST(alpha = 0.05, logscale = TRUE,
                    theta0 = Ratio_AUC, theta1 = 0.80,
                    theta2=1.25, n=c(n1, n2),
                    CV=(CVintra_AUC/100),
                    design = "2x2x4") # Calculate of Power TOST for AUC
```

```
# Model of Cmáx for to meet CI 90%, CVintra and Power TOST

Modelo_Cmax_1 = lme(log(Cmax) ~ TRT + GRP + PRD, random=~1|SUBJ, data=PK) # Use of function lme
varia_Cmax = data.frame(VarCorr(Modelo_Cmax_1)) # Matrix of Variability
CVinter_Cmax = (sqrt(exp(as.numeric(varia_Cmax[1,1][1]))-1))*100 # Calculate of CVinter
CVintra_Cmax = (sqrt(exp(as.numeric(varia_Cmax[2,1][1]))-1))*100 # Calculate of CVintra
Type_III_Cmax = anova(Modelo_Cmax_1) # Calculate of efects
attr(Type_III_Cmax, "heading")[1] = "Type III Analysis of Variance Table\n"
ci_Cmax = intervals(Modelo_Cmax_1, 0.90, which="fixed") # CI 90%

RxT_Cmax = data.frame(Estimativa = round(exp(ci_Cmax$fixed["TRTT",]),5)) # Calculate Ratio and CI 90%

Ratio_Cmax = as.numeric(RxT_Cmax[2,1]) # Ratio of Cmáx

Poder_Cmax = 100*power.TOST(alpha = 0.05, logscale = TRUE,
                    theta0 = Ratio_Cmax, theta1 = 0.80,
                    theta2=1.25, n=c(n1,n2), CV=(CVintra_Cmax/100),
                    design = "2x2x4") # Calculate of Power TOST for Cmax

# Table of Results with with simple size, Ratio, CI 90%, CVinter, CVintra and Power TOST

result = data.frame(PK = c("Cmax", "AUC"),
                n_A = c(n1, n2),
                n_B = c(n1, n2),
                Ratio = c(Ratio_Cmax*100, Ratio_AUC*100),
                LL = c(RxT_Cmax[1,1]*100, RxT_AUC[1,1]*100),
                UL = c(RxT_Cmax[3,1]*100, RxT_AUC[3,1]*100),
                CV_Between = c(CVinter_Cmax, CVinter_AUC),
                CV_Within = c(CVintra_Cmax, CVintra_AUC),
                Power_TOST = c(Poder_Cmax, Poder_AUC))

# Analysis of Variability for to calculate CVwr and CVwt #

# For ASC0-t #

dat_Ref_ASC = PK %>% filter(TRT == "R") # Separating of date Reference
model_CVwr_ASC  <- lm(log(AUClast) ~ GRP + SUBJ %in% SUBJ + PRD, dat_Ref_ASC) # Models only with ASC Reference data
CVwr_ASC  = round(sqrt(exp(anova(model_CVwr_ASC)["Residuals", "Mean Sq"]) - 1)*100, 3) # Estimate of CVwr of ASC - Reference

dat_Test_ASC  = PK %>% filter(TRT == "T") # Separating of date Test
```

```
  model_CVwt_ASC  <- lm(log(AUClast) ~ GRP + SUBJ %in% SUBJ + PRD, dat_Test_ASC) # Models only with ASC Test data
  CVwt_ASC  = round(sqrt(exp(anova(model_CVwt_ASC)["Residuals", "Mean Sq"]) - 1)*100, 3) # Estimate of CVwr of ASC - Test

  # For Cmax #

  dat_Ref_Cmax = PK %>% filter(TRT == "R") # Separating of date Reference
  model_CVwr_Cmax <- lm(log(Cmax) ~ GRP + SUBJ %in% SUBJ + PRD, dat_Ref_Cmax) # Models only with Cmax Reference data
  CVwr_Cmax = round(sqrt(exp(anova(model_CVwr_Cmax)["Residuals", "Mean Sq"]) - 1)*100, 3) # Estimate of CVwr of Cmax - Reference

  dat_Test_Cmax = PK %>% filter(TRT == "T") # Separating of date Test
  model_CVwt_Cmax <- lm(log(Cmax) ~ GRP + SUBJ %in% SUBJ + PRD, dat_Test_Cmax) # Models only with Cmax Test data
  CVwt_Cmax = round(sqrt(exp(anova(model_CVwt_Cmax)["Residuals", "Mean Sq"]) - 1)*100, 3) # Estimate of CVwr of Cmax - Test

  result1 = result %>% mutate(CVwr = c(CVwr_Cmax, CVwr_ASC), CVwt = c(CVwt_Cmax, CVwt_ASC)) # Results of analysis of variability

  return(result1)

 }else{
  print("unknown design")  # For different cases of desgin studies "2x2x2", "2x3x3" and "2x2x4"
 }

}
```

Applying the function in the NCAResult4BE data of the BE pac...

6 Command:

```
write.table(NCAResult4BE, "dat_test.csv", row.names = FALSE, sep = ";", dec = ",")  # Saving data in the directory
dat_test = read.table("dat_test.csv", header = T, sep = ";", dec = ",") # Loading date
be2x2(dat_test, c("AUClast", "Cmax", "Tmax")) # Applying the be_2x2() function of the "BE" package
bioequivalencia_f2_V1(dat_test, "2x2") # Applying the function bioequivalencia_f2_V1()
```