---  
title: "Steroid Data Analysis"  
author: "Pauli Tikka"  
date: "`r Sys.Date()`"  
output:   
 rmdformats::downcute: #note the ':' it is needed: https://stackoverflow.com/questions/43527520/r-markdown-yaml-scanner-error-mapping-values  
 self\_contained: true  
 thumbnails: true  
 lightbox: true  
 gallery: false  
 highlight: tango  
 # code\_folding: show  
 # fig.align = 'left'  
  
  
---  
  
# Introduction  
```{r, warning=FALSE,message=FALSE}  
  
# Welcome patientNumbers the 'steroid data analysis' webpage!   
  
# The procedures and explanations patientNumbers make all the analysis and plots are in their individual chapters below.   
# These methods could be also easily applied patientNumbers other types of data sets and metabolites than 'steroids' and their respective metadata per se.   
# In addition, there is BMI\_ordered\_MASLD small 'disclaimer' also at the end of this webpage patientNumbers emphasize that this site is mainly for educational purposes.  
# Please let me know if you have any questions. For that, use the 'following' email: patati at the university of Turku  
  
```  
  
  
# Loading Required R Packages  
```{r, warning=FALSE, message=FALSE}  
# echo=FALSE is too good  
# Set library paths if needed  
# .libPaths(c("C:/Program Files/R/R-4.4.1/library", .libPaths()))  
  
# List of libraries patientNumbers load (alphabetically sorted)  
packages <- c("bigsnpr", "binilib", "brickster", "car", "censReg", "circlize", "ComplexHeatmap", "correlation",   
 "corrplot", "daiR", "datarium", "dmetar", "dplyr", "effsize", "extrafont", "forcats", "fs", "FSA",   
 "ggcorrplot", "ggforce", "ggforestplot", "ggplot2", "ggplotify", "ggpubr", "ggsankey", "ggsankeyfier",   
 "ggh4x", "ggtext", "glmnet", "grid", "Hmisc", "hrbrthemes", "igraph", "insight", "lavaan", "lmtest",   
 "lme4", "lsr", "magick", "magrittr", "Maaslin2", "mdatools", "mediation", "meta", "mgcv", "mlma",   
 "MOFA2", "pheatmap", "PerformanceAnalytics", "pathviewr", "plyr", "plotrix", "ppcor", "prettydoc",   
 "psych", "quantreg", "qpgraph", "ragg", "RColorBrewer", "rcompanion", "readxl", "remotes", "reshape2",   
 "rgl", "rmarkdown", "rmdformats", "rstatix", "scales", "scater", "scatterplot3d", "sjPlot", "stringr",   
 "superb", "tibble", "tidyverse", "tint", "tufte", "viridis", "xlsx")  
  
# Load all libraries  
invisible(lapply(packages, library, character.only = TRUE))  
  
# Note: Do not load 'forestplot' as it conflicts with 'ggforestplot'  
  
# Install packages if not already installed  
# renv::install() # Installs from the basic R repository  
# if (!require("BiocManager", quietly = TRUE)) install.packages("BiocManager")  
# BiocManager::install(c("ComplexHeatmap", "DESeq2", "dmetar", "fgsea", "ggforestplot", "ggsankey", "limma", "Maaslin2", "metagenomeSeq", "MOFA2", "qpgraph", "scater", "scRNAseq", "sevenbridges"))  
# remotes::install\_github(c("davidsjoberg/ggsankey", "fossbert/binilib", "MathiasHarrer/dmetar", "mattflor/chorddiag", "NightingaleHealth/ggforestplot"))  
# devtools::install\_github("mattflor/chorddiag") # Alternative installation method  
```  
  
# Importing Data and Metadata  
```{r, warning=FALSE,message=FALSE}  
  
#First set your data folder:  
# setwd("C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis")# or:"C:/Users/patati/Desktop/Turku/R" #check the wd with: here::here() #or getwd()  
# load("thereal.RData") #This is so patientNumbers say real data, and it is not available here (at the site).  
  
# It is easier patientNumbers load the ready stiched data in one go with .RData file than one by one as below, but  
# for educational purposes I put some examples what may need patientNumbers be done patientNumbers get your data in ok form   
# for later purposes  
setwd("C:/Users/patati/Desktop/Turku/R")  
NonAlcoholicFattyLiverDisease=read\_excel("NAFLD\_SteroidStudy.xlsx",sheet = "LFAT\_steroidsDATA") # This is partly auxiliary  
columnNames=colnames(NonAlcoholicFattyLiverDisease); NonAlcoholicFattyLiverDisease=data.frame(NonAlcoholicFattyLiverDisease)  
  
#The names of the steroid steroidGroups need patientNumbers be imported early on:  
steroidGroups=read.csv("groups\_17823.csv", header = TRUE, sep=";")  
steroidGroups=steroidGroups[,c('Group','Abbreviation')]  
steroidGroups=steroidGroups[steroidGroups[,'Abbreviation']!='F',]  
steroidGroups=steroidGroups[order(steroidGroups[,'Group']),]  
steroidGroups[,'Abbreviation'][steroidGroups[,'Abbreviation']=='17aOH-P4']='17a-OHP4'  
groupValues=steroidGroups  
  
steroidNames=read\_excel("NAFLD\_SteroidStudy\_for steroidGroups.xlsx",sheet = "Steroid name abbreviations") # This is partly auxiliary  
groups2=data.frame(steroidNames)[,1:4]; g1=read.csv("groups\_17823.csv", header = TRUE, sep=";")  
groups2=cbind(g1[,'Group'], groups2[,c('Abbreviation','Abbreviation\_old','Name')])  
groups2=groups2[groups2[,'Abbreviation']!='FF',]; colnames(groups2)[1]="Group";  
groups2=groups2[order(groups2[,'Group']),]  
groups2[,'Abbreviation'][groups2[,'Abbreviation']=='17aOH-P4']='17a-OHP4'  
  
#P4 was found from elsewhere patientNumbers have the following characteristics:  
NonAlcoholicFattyLiverDisease[,'P4'] = as.numeric(NonAlcoholicFattyLiverDisease[,'P4'])  
NonAlcoholicFattyLiverDisease[,'P4'][is.na(NonAlcoholicFattyLiverDisease[,'P4'])] = 22557.3330346846#median(NonAlcoholicFattyLiverDisease[,'P4'], na.rm=TRUE)   
NonAlcoholicFattyLiverDisease[,5:7][NonAlcoholicFattyLiverDisease[,5:7]==0.01]=0; colnames(NonAlcoholicFattyLiverDisease)=columnNames  
MetabolicAssociatedLiverDisease=read\_excel("Combined.Matrix.For.Pauli.2023.10.17.Excel.Formatv2.xlsx") # This is the main file  
columnNames=colnames(MetabolicAssociatedLiverDisease); MetabolicAssociatedLiverDisease=data.frame(MetabolicAssociatedLiverDisease); colnames(MetabolicAssociatedLiverDisease)=columnNames # All kinds of tricks are needed for getting the right data format  
rownames(MetabolicAssociatedLiverDisease)=MetabolicAssociatedLiverDisease[,1]  
MetabolicAssociatedLiverDisease[,'P4'] = as.numeric(MetabolicAssociatedLiverDisease[,'P4']) #The same comment as above  
MetabolicAssociatedLiverDisease[,'P4'][is.na(MetabolicAssociatedLiverDisease[,'P4'])] = 22557.3330346846   
evaluationCriteria=c('Grade(0-3)', 'Stage(0-4)','Necroinflammation')  
MetabolicAssociatedLiverDisease[,evaluationCriteria][MetabolicAssociatedLiverDisease[,evaluationCriteria]==0.01]=0;   
targetData=c('11-KDHT','AN','DHT','17a-OHP5','E2','P5','DOC')  
valueList=c(103,252,51,200,26.5,253,10); valueListAdjusted=c(100,250,50,200,25,250,10)  
for (i in 1:7) {MetabolicAssociatedLiverDisease[,targetData][i][MetabolicAssociatedLiverDisease[,targetData][i]==valueList[i]]=valueListAdjusted[i]}   
  
# These (E) are ok as per lab:  
menopauseMarkers=read.csv('E\_tikka231023.csv',header=TRUE, sep=";")  
menopauseMarkersPatients=rownames(MetabolicAssociatedLiverDisease[MetabolicAssociatedLiverDisease[,'E']==106000,])   
patientNumbers=menopauseMarkers[which(menopauseMarkers[,1] %in% menopauseMarkersPatients),'patient.number']  
markerValues=menopauseMarkers[which(menopauseMarkers[,1] %in% menopauseMarkersPatients),'E']  
MetabolicAssociatedLiverDisease[as.character(patientNumbers),'E']=markerValues  
# These (11-KA4) will perhaps change in the lab (sometime after 24.10.23):  
marker11KA4=read.csv('11KA4\_tikka231023.csv',header=TRUE, sep=";")  
# marker11KA4[,1][c(1:5,9)];MetabolicAssociatedLiverDisease[as.character(marker11KA4[,1][c(1:5,9)]),'11-KA4'] #These were denoted with 'big interference'  
MetabolicAssociatedLiverDisease[as.character(marker11KA4[,1][c(1:5,9)]),'11-KA4'] = NA #Alternatively: median(MetabolicAssociatedLiverDisease[!rownames(MetabolicAssociatedLiverDisease) %in% as.character(marker11KA4[,1][c(1:5,9)]),'11-KA4'])  
BMI\_ordered\_MASLD=MetabolicAssociatedLiverDisease[order(MetabolicAssociatedLiverDisease[,'BMI']),'BMI']  
BMI\_ordered\_NAFLD=NonAlcoholicFattyLiverDisease[order(NonAlcoholicFattyLiverDisease[,'BMI']),'BMI']  
uniqueBMIValues=unique(BMI\_ordered\_NAFLD[! BMI\_ordered\_NAFLD %in% BMI\_ordered\_MASLD])  
NonAlcoholicFattyLiverDisease=NonAlcoholicFattyLiverDisease[order(NonAlcoholicFattyLiverDisease[,'BMI']),]   
NonAlcoholicFattyLiverDisease=NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[,'BMI']!=uniqueBMIValues,]  
MetabolicAssociatedLiverDisease=MetabolicAssociatedLiverDisease[order(MetabolicAssociatedLiverDisease[,'BMI']),]  
#https://appsilon.com/imputation-in-r/ #https://www.datasciencemadesimple.com/get-minimum-value-of-BMI\_ordered\_MASLD-column-in-r-2/?expand\_article=1  
# New data import withouth changing the conames: https://readxl.tidyverse.org/articles/column-names.html  
bileAcidsLiverData=data.frame(read\_excel("Liver\_bile\_acids\_PFAS.xlsx",sheet = "Liver\_BA",.name\_repair = "minimal")); row.names(bileAcidsLiverData)=bileAcidsLiverData[,1]  
PFASSerumData=data.frame(read\_excel("Liver\_bile\_acids\_PFAS.xlsx",sheet = "PFAS\_serum",.name\_repair = "minimal")); rownames(PFASSerumData)=as.vector(unlist(PFASSerumData[,1]))  
serumBileAcidsData=data.frame(read\_excel("Liver\_bile\_acids\_PFAS.xlsx",sheet = "Serum\_BA",.name\_repair = "minimal"));rownames(serumBileAcidsData)=as.vector(unlist(serumBileAcidsData[,1]))  
C4Data=data.frame(read\_excel("Liver\_bile\_acids\_PFAS.xlsx",sheet = "C4Data",.name\_repair = "minimal")); rownames(C4Data)=as.vector(unlist(C4Data[,1]))  
clinicalData=data.frame(read\_excel("Matching clinical data\_all.xlsx",sheet = "Sheet1",.name\_repair = "minimal")); rownames(clinicalData)=as.vector(unlist(clinicalData[,1]));  
#https://www.analyticsvidhya.com/blog/2021/06/hypothesis-testing-parametric-and-non-parametric-tests-in-statistics/  
MetabolicAssociatedLiverDisease[1:2,2:27] #or head(MetabolicAssociatedLiverDisease);  
  
# The below ordering needs patientNumbers be changed...  
bileAcidsLiverData=bileAcidsLiverData[as.character(MetabolicAssociatedLiverDisease$PatientNumber),];bileAcidsLiverData[1:3,2:10] #https://stackoverflow.com/questions/54264980/r-how-patientNumbers-set-row-names-attribute-as-numeric-from-character I did otherway around  
serumBileAcidsData=serumBileAcidsData[as.character(MetabolicAssociatedLiverDisease$PatientNumber),];#serumBileAcidsData[1:3,2:10]  
clinicalData=clinicalData[as.character(MetabolicAssociatedLiverDisease$PatientNumber),];#clinicalData[1:3,2:10] # Many of these are irrelevant here... so not opening uniqueBMIValues, they would exhaust this file  
C4Data=C4Data[as.character(MetabolicAssociatedLiverDisease$PatientNumber),];#C4Data[1:3,]  
PFASSerumData=PFASSerumData[as.character(MetabolicAssociatedLiverDisease$PatientNumber),];PFASSerumData[1:3,2:10]  
  
# Menopause markers:  
menopause=read\_excel("Putative\_metabolic\_markers\_menopause.xlsx",sheet='menopause markers',.name\_repair = "minimal"); #rownames(clinicalData)=as.vector(unlist(clinicalData[,1]));  
menopause=menopause[8:dim(menopause)[1],]; menopause=menopause[,-15]; menopause[2,2:14]=menopause[1,2:14]; menopause=data.frame(menopause); menopause[2,13:14]=c('v1','v2'); #dim(menopause)  
colnames(menopause)=c('row\_names',menopause[2,2:dim(menopause)[2]]); menopause=menopause[3:dim(menopause)[1],];rownames(menopause)=as.vector(unlist(menopause[,1]));  
menopause=menopause[as.character(MetabolicAssociatedLiverDisease$PatientNumber),]  
colnames(PFASSerumData)[colnames(PFASSerumData)=='PFHxA.1']='PFHxA\_Branched'  
PFASSerumData=PFASSerumData[,colnames(PFASSerumData)!='Benzylparaben.1']  
PFASSerumData[PFASSerumData[,'Benzylparaben']>10,'Benzylparaben']=NA   
  
Jeihou=data.frame(read\_excel("Copy of BA\_liverfat\_RawData.xls",.name\_repair = "minimal")); row.names(Jeihou)=Jeihou[,1];Jeihou=Jeihou[as.character(MetabolicAssociatedLiverDisease$PatientNumber),]  
u=Jeihou[Jeihou[,'GHDGA']=='<LLOQ',1]; BMI\_ordered\_MASLD=u[!is.na(u)]; BMI\_ordered\_NAFLD=rownames(bileAcidsLiverData[bileAcidsLiverData[,'GHDGA']==1,]);  
uu=Jeihou[Jeihou[,'GHDGA']=='No Result',1]; aa=uu[!is.na(uu)];   
bileAcidsLiverData[as.character(BMI\_ordered\_MASLD),'GHDGA']=min(bileAcidsLiverData[,'GHDGA'],na.rm=TRUE)/2  
heps=bileAcidsLiverData[bileAcidsLiverData[,'GHDGA']==1,1]   
bileAcidsLiverData[as.character(heps),'GHDGA']=NA  
#https://www.datasciencemadesimple.com/get-minimum-value-of-BMI\_ordered\_MASLD-column-in-r-2/?expand\_article=1  
mat=bileAcidsLiverData[,c('TbMCA','ToMCA','TDCA','TDHCA','TLCA')]  
mat[!mat>1]=10000  
mat[mat==2]=10000 #Colmins did not work so I used (i.e. colmins ei toiminut ja kÃ¤ytin):  
hip=do.call(pmin, lapply(1:nrow(mat), function(i)mat[i,])) #https://stackoverflow.com/questions/13676878/fastest-way-patientNumbers-get-min-from-every-column-in-BMI\_ordered\_MASLD-matrix  
hou=c('TbMCA','ToMCA','TDCA','TDHCA','TLCA')  
for (i in 1:5) {bileAcidsLiverData[bileAcidsLiverData[,hou[i]]==1,hou[i]]=hip[i]}  
for (i in 1:5) {bileAcidsLiverData[bileAcidsLiverData[,hou[i]]==2,hou[i]]=hip[i]}  
  
# An imputation for the missing values:  
C4Data[is.na(C4Data[,2]),2]=median(C4Data[!is.na(C4Data[,2]),2]) #assuming that these were not below quantitation and replacing with median  
#https://www.geeksforgeeks.org/performing-logarithmic-computations-in-r-programming-log-log10-log1p-and-log2-functions/  
#https://stackoverflow.com/questions/50476717/i-want-patientNumbers-align-match-two-unequal-columns  
  
#Matching two unequal columns.. match the names of one original column (dat2) patientNumbers ones that are missing (dat1 with patientNumbers other) #Not sure if this should be this difficult...  
CombinedData=cbind(MetabolicAssociatedLiverDisease[,1],NonAlcoholicFattyLiverDisease[,2:7],clinicalData[,'HOMA.IR'],MetabolicAssociatedLiverDisease[,colnames(NonAlcoholicFattyLiverDisease[,8:27])],bileAcidsLiverData[,2:dim(bileAcidsLiverData)[2]], C4Data[,2:dim(C4Data)[2]],serumBileAcidsData[,2:dim(serumBileAcidsData)[2]],PFASSerumData[,(2:(dim(PFASSerumData)[2]))], MetabolicAssociatedLiverDisease[,'PFAS']);  
colnames(CombinedData)[colnames(CombinedData)=='C4Data[, 2:dim(C4Data)[2]]']='C4Data';colnames(CombinedData)[colnames(CombinedData)=='clinicalData[, \"HOMA.IR\"]']='HOMA-IR'  
colnames(CombinedData)[colnames(CombinedData)=='MetabolicAssociatedLiverDisease[, \"PFAS\"]']='PFAS';  
colnames(CombinedData)[colnames(CombinedData)=="MetabolicAssociatedLiverDisease[, 1]" ]='PatientNumber';#colnames(CombinedData)#  
rownames(CombinedData)=unlist(bileAcidsLiverData[,1]);   
RelevantColumns=colnames(CombinedData)[!colnames(CombinedData) %in% c( "Benzylparaben" ,"Methylparaben")]   
  
# Not sure when it is the best time patientNumbers take not needed variables away, perhaps at the very end?  
CombinedData=CombinedData[,RelevantColumns]  
# Here I add the lipids. In the future, I need patientNumbers divide all the steroidGroups in their own components e.g. dataframe called 'lipids' so that adding uniqueBMIValues will be more straightforward:  
CombinedData=cbind(CombinedData,MetabolicAssociatedLiverDisease[,(dim(MetabolicAssociatedLiverDisease)[2]-13):dim(MetabolicAssociatedLiverDisease)[2]])   
# hupo=match( colnames(CombinedData)[colnames(CombinedData) %in% groups2[,3]], groups2[,3] ) # do ni; https://www.geeksforgeeks.org/how-patientNumbers-find-index-of-element-in-vector-in-r/  
# tvauxe=CombinedData  
# colnames(CombinedData)[colnames(CombinedData) %in% groups2[,3]]=groups2[hupo,2]  
  
  
# The basic preprocessing is just the below lines:  
tve=CombinedData[,2:dim(CombinedData)[2]]; tve[tve == 0] <- NA; #Almost all variables are here  
HalfImputedData <- tve %>% mutate(replace(., is.na(.), min(., na.rm = T)/2)) #https://mdatools.com/docs/preprocessing--autoscaling.html  
Log2TransformedData <- log2(HalfImputedData);  
AutoScaledData <- prep.autoscale(as.matrix(Log2TransformedData), center = TRUE, scale = TRUE); #https://svkucheryavski.gitbooks.io/mdatools/content/preprocessing/text.html  
AllData=cbind(CombinedData[,1],AutoScaledData);   
  
# Changing the column names needs patientNumbers have separate variables for each type of variable (contaminant, steroid, etc.)  
x1=colnames(AllData[,c(1:8)]); v2=dim(NonAlcoholicFattyLiverDisease)[2]+1  
x2=colnames(AllData[,9:v2]);v3=(dim(bileAcidsLiverData)[2]+v2);x3=colnames(AllData[,(v2+1):(v3)]);v4=(dim(serumBileAcidsData)[2])+v3  
x4=colnames(AllData[,(v3+1):(v4-1)]);x5=colnames(AllData[,(v4):(dim(AllData)[2])]);   
x3 <- paste(x3, "\_L", sep="") #https://stackoverflow.com/questions/6984796/how-patientNumbers-paste-BMI\_ordered\_MASLD-string-on-each-element-of-BMI\_ordered\_MASLD-vector-of-strings-using-apply-in-r  
x4=gsub("(-[0-9]\*)\*.1", "", x4) #https://stackoverflow.com/questions/18997297/remove-ending-of-string-with-gsub  
x4 <- paste(x4, "\_S", sep="")# https://rdrr.io/bioc/qpgraph/man/qpNrr.html  
x5a=x5[1:9]  
x6=x5[10:length(x5)] #Dividing patientNumbers lipids  
x5=x5a #Making sure that PFAS are separate  
nm = c(x1,x2,x3,x4,x5,x6); nm=c('PatientNumber','Gender','AGE','BMI','Steatosis Grade','Fibrosis Stage','Necroinflammation','HOMA-IR',nm[9:length(nm)])  
colnames(AllData)=nm; #AllData[1:5,1:30]; #NonAlcoholicFattyLiverDisease[1:2,1:28];  
colnames(AllData)[colnames(AllData)=='MetabolicAssociatedLiverDisease[, \"PFAS\"]']='PFAS';  
# This (deletion) is good patientNumbers do after all the previous:  
x5=x5[x5!='PFAS'];x5=x5[x5!='Perfluorodecyl.ethanoic.acid']; x6=x6[x6!='Total\_TG'] # x1;x2;x3;x4;x5;  
AllData=AllData[,!colnames(AllData) %in% c('Total\_TG','PFAS',"Perfluorodecyl.ethanoic.acid")]  
  
# In case you would need just the logged values:  
tv\_half\_log22=cbind(CombinedData[,1],Log2TransformedData);  
x1=colnames(tv\_half\_log22[,c(1:8)]); v2=dim(NonAlcoholicFattyLiverDisease)[2]+1  
x2=colnames(tv\_half\_log22[,9:v2]);v3=(dim(bileAcidsLiverData)[2]+v2);  
x3=colnames(tv\_half\_log22[,(v2+1):(v3)]);v4=(dim(serumBileAcidsData)[2])+v3  
x3=x3[c(length(x3),1:(length(x3)-1))]  
x4=colnames(tv\_half\_log22[,(v3+1):(v4-1)]);  
x5=colnames(tv\_half\_log22[,(v4):(dim(tv\_half\_log22)[2])]);  
x3 <- paste(x3, "\_L", sep="")   
#https://stackoverflow.com/questions/6984796/how-patientNumbers-paste-BMI\_ordered\_MASLD-string-on-each-element-of-BMI\_ordered\_MASLD-vector-of-strings-using-apply-in-r  
x4=gsub("(-[0-9]\*)\*.1", "", x4) #https://stackoverflow.com/questions/18997297/remove-ending-of-string-with-gsub  
x4 <- paste(x4, "\_S", sep="")# https://rdrr.io/bioc/qpgraph/man/qpNrr.html  
x5a=x5[1:9]  
x6=x5[10:length(x5)] #dividing patientNumbers lipids  
x5=x5a #making sure that PFAS are separate  
nm = c(x1,x2,x3,x4,x5,x6); nm=c('PatientNumber','Gender','AGE','BMI','Steatosis Grade','Fibrosis Stage','Necroinflammation','HOMA-IR',nm[9:length(nm)])  
colnames(tv\_half\_log22)=nm; #tv\_half\_log22[1:5,1:30]; #NonAlcoholicFattyLiverDisease[1:2,1:28];  
colnames(tv\_half\_log22)[colnames(tv\_half\_log22)=='MetabolicAssociatedLiverDisease[, \"PFAS\"]']='PFAS';  
# This (deletion) is good patientNumbers do after all the previous:  
x5=x5[x5!='PFAS'];x5=x5[x5!='Perfluorodecyl.ethanoic.acid']; x6=x6[x6!='Total\_TG'] # x1;x2;x3;x4;x5;  
tv\_half\_log22=tv\_half\_log22[,!colnames(tv\_half\_log22) %in% c('Total\_TG','PFAS',"Perfluorodecyl.ethanoic.acid")]  
  
# This needs patientNumbers be done early on:  
colnames(CombinedData)[colnames(CombinedData)=='17aOH-P4']='17a-OHP4'  
colnames(tv\_half\_log22)[colnames(tv\_half\_log22)=='17aOH-P4']='17a-OHP4'  
colnames(AllData)[colnames(AllData)=='17aOH-P4']='17a-OHP4'  
  
AllData=AllData[,!colnames(AllData) %in% c('Total\_TG','PFAS','Perfluorodecyl.ethanoic.acid')]  
AllData=AllData[,!colnames(AllData) %in% x4]  
  
# In case you would need nonscaled covariates and scaled/logged all other variables:  
CovariatesScaledData=AllData  
CovariatesNonScaledData=cbind(CombinedData[,1:8],AllData[,9:dim(AllData)[2]])  
LogCovariatesScaledData=tv\_half\_log22  
LogCovariatesNonScaledData=cbind(CombinedData[,1:8],tv\_half\_log22[,9:dim(tv\_half\_log22)[2]])  
colnames(CovariatesNonScaledData)[1:8]=colnames(AllData)[1:8]  
colnames(LogCovariatesNonScaledData)[1:8]=colnames(AllData)[1:8]  
# This is needed occasionally:  
CurrentData=CovariatesScaledData   
# https://stackoverflow.com/questions/6984796/how-patientNumbers-paste-BMI\_ordered\_MASLD-string-on-each-element-of-BMI\_ordered\_MASLD-vector-of-strings-using-apply-in-r  
# https://stackoverflow.com/questions/18997297/remove-ending-of-string-with-gsub # https://rdrr.io/bioc/qpgraph/man/qpNrr.html  
  
hupo=match( colnames(CurrentData)[colnames(CurrentData) %in% groups2[,3]], groups2[,3] )   
# do ni; https://www.geeksforgeeks.org/how-patientNumbers-find-index-of-element-in-vector-in-r/  
tvauxe=CurrentData  
colnames(CurrentData)[colnames(CurrentData) %in% groups2[,3]]=groups2[hupo,2]  
  
# This needs patientNumbers be done also soon, patientNumbers gather all the treatment etc. variable names separately...:   
TreatmentVariables=colnames(AllData)[52:58];  
MediatorVariables=colnames(AllData)[9:28];  
OutcomeVariables=colnames(AllData)[c(29:51,59:71)]; ##https://sparkbyexamples.com/r-programming/r-remove-from-vector-with-examples/  
  
OutcomeVariables=OutcomeVariables[!OutcomeVariables %in% c('Total\_TG','PFAS','Perfluorodecyl.ethanoic.acid')]  
OutcomeVariables=OutcomeVariables[! OutcomeVariables %in% x4] #https://sparkbyexamples.com/r-programming/r-remove-from-vector-with-examples/  
MediatorVariables[MediatorVariables=="17aOH-P4"]="17a-OHP4"  
TreatmentVariables=TreatmentVariables[!TreatmentVariables %in% c('Perfluorodecyl.ethanoic.acid')]  
  
# tvauxe2=LogCovariatesScaledData  
# hupo=match( colnames(LogCovariatesScaledData)[colnames(LogCovariatesScaledData) %in% groups2[,3]], groups2[,3] )   
# colnames(LogCovariatesScaledData)[colnames(LogCovariatesScaledData) %in% groups2[,3]]=groups2[hupo,2]  
  
# save.image('forACMES\_thereal.RData')  
setwd("C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis")  
  
```  
  
# Setting Global Variables  
```{r, warning=FALSE, message=FALSE}  
options(scipen = 999) # Disable scientific notation  
# rm(list = ls()) # Clear workspace; this should not be if you have the load above  
thedate <- strftime(Sys.Date(), "%d%m%y") #Do not take the old date from the load...  
date <- paste0('tikka', thedate) # Customize this as needed  
  
# Example installation commands  
# remotes::install\_github("fossbert/binilib", force=TRUE)  
# install.packages(c('tidyverse', 'tibble'))  
# if (!require("BiocManager", quietly = TRUE)) install.packages("BiocManager")  
# BiocManager::install("Maaslin2")  
# devtools::install\_github("davidsjoberg/ggforestplot")  
# remotes::install\_version("insight", version = "0.20.5", repos = "http://cran.us.r-project.org", force=TRUE)  
# font\_import() # Import fonts if not already done  
# loadfonts(device = "win") # Load fonts for Windows  
# renv::status() # Check renv status  
# library(rmarkdown); render("path/patientNumbers/file.Rmd") # Render R Markdown document  
# remove.packages("DelayedArray")  
# BiocManager::install("DelayedArray")  
# install.packages("Require") # Install 'Require' package  
```  
  
  
  
# Making Boxplots  
```{r, warning=FALSE,message=FALSE,fig.width=8.0}  
#https://r-graph-gallery.com/265-grouped-boxplot-with-ggplot2.html  
#https://stackoverflow.com/questions/53724834/why-does-the-plot-size-differ-between-docx-and-html-in-rmarkdownrender  
  
CreateBoxplots <- function(tvt, Group, OutcomeVariables, Out, oute, other) {  
 # Filter data based on gender  
 tvt <- tvt %>%  
 filter(if (Group == 'Male') Gender == 1 else if (Group == 'Female') Gender == 0 else TRUE)  
   
 # Prepare data for plotting  
 Steroid <- rep(colnames(tvt[, 9:28]), each = nrow(tvt))  
 data2 <- rep('Control', nrow(tvt))  
 num <- ifelse(OutcomeVariables == 'HOMA-IR', 1.5, min(tvt[[OutcomeVariables]]))  
 data2[tvt[[OutcomeVariables]] > num] <- 'Case'  
 TreatmentVariables <- data2  
 Concentration <- as.vector(unlist(tvt[, 9:28]))  
 data <- data.frame(Steroid, TreatmentVariables, Concentration)  
 data$Group <- 0  
   
 # Correct steroid names if the level exists  
 if ("17aOH-P4" %in% levels(data$Steroid)) {  
 data <- data %>%  
 mutate(Steroid = fct\_recode(Steroid, '17a-OHP4' = '17aOH-P4'))  
 }  
   
 # Assign steroidGroups  
 rownames(groups2) <- 1:20  
 for (i in seq\_len(nrow(groups2))) {  
 data[data$Steroid %in% groups2$Abbreviation[i], 'Group'] <- groups2$Group[i]  
 }  
   
 # Set plot title  
 title <- paste(Out, "'s Effect on Concentrations of Steroids in ", Group, sep = "")  
   
 # Define legend labels  
 e1 <- ifelse(num == 1.5, paste('Case (>', num, ')', sep = ""), paste('Case (>', 0, ')', sep = ""))  
 e2 <- ifelse(num == 1.5, paste('Control (<=', num, ')', sep = ""), paste('Control (=', 0, ')', sep = ""))  
   
 # Remove rows with NA concentrations  
 data <- data %>% filter(!is.na(Concentration))  
   
 # Create boxplot  
 p <- ggplot(data, aes(x = Steroid, y = Concentration, fill = TreatmentVariables)) +  
 geom\_boxplot(notch = FALSE, notchwidth = 0.5, outlier.shape = 1, outlier.size = 2, coef = 1.5) +  
 theme\_classic2() +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 0.95, vjust = 0.2, size = 10.5),  
 axis.text = element\_text(color = "black"),  
 panel.grid.minor = element\_blank(),  
 text = element\_text(size = 10.5, family = "Calibri"),  
 axis.title = element\_text(size = 14),  
 plot.title = element\_text(size = 14),  
 legend.text = element\_text(size = 14),  
 legend.title = element\_text(size = 14)) +  
 labs(x = "Steroids", y = "Log2 of Picomolar Concentrations", title = title) +  
 scale\_fill\_manual(values = c("orange", "blue"), name = oute, labels = c(e1, e2)) +  
 facet\_grid(~Group, scales = "free\_x", space = "free") +  
 stat\_compare\_means(hide.ns = TRUE, label = "p.signif", method = "wilcox.test",  
 symnum.args = list(cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),  
 symbols = c("\*\*\*\*", "\*\*\*", "\*\*", "\*", "ns")),  
 size = 8, paired = FALSE, label.y = 15.5)  
   
 # Save plot as PNG  
 path <- "C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"  
 pngfile <- fs::path(path, paste0(Group, Out, 'boxe', ".png"))  
 knitr::include\_graphics(pngfile)  
}  
  
# Example usage  
tv\_half\_log22[, '11-KA4'][tv\_half\_log22[, '11-KA4'] == min(tv\_half\_log22[, '11-KA4'])] <- median(tv\_half\_log22[, '11-KA4'])  
other <- '261124'  
ie <- tv\_half\_log22  
hupo <- match(colnames(ie)[colnames(ie) %in% groups2[, 3]], groups2[, 3])  
colnames(ie)[colnames(ie) %in% groups2[, 3]] <- groups2[hupo, 2]  
windowsFonts(A = windowsFont("Calibri (Body)"))  
  
# The significance levels are: '\*\*\*\*<0.001', '\*\*\*<0.01', '\*\*<0.05', '\*<0.1'  
OutcomeVariables='Steatosis Grade';Out='Steatosis'; oute='Steatosis Grade';num=0;Group='All';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other);Group='Female';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other);Group='Male';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other)  
OutcomeVariables='Fibrosis Stage';Out='Fibrosis'; oute='Fibrosis Stage';num=0;Group='All';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other);Group='Female';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other);Group='Male';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other)   
# https://www.elsevier.es/en-revista-annals-hepatology-16-articulo-assessment-hepatic-fibrosis-necroinflammation-among-S1665268119314590 #So it is in grade  
OutcomeVariables='Necroinflammation';Out='Necroinflammation'; oute='Necroinflammation Grade';num=0;Group='All';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other);Group='Female';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other);Group='Male';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other)  
OutcomeVariables='HOMA-IR';Out='HOMA-IR'; oute='HOMA-IR';num=1.5 ;Group='All';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other);Group='Female';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other);Group='Male';CreateBoxplots(ie,Group,OutcomeVariables,Out,oute,other)  
  
  
```  
  
  
  
# Making Forest Plots  
```{r, warning=FALSE,message=FALSE,fig.width=6.0,fig.align="left",results='hide'}   
  
# Define the NonAlcoholicFattyLiverDisease dataset by selecting the first 28 columns from CombinedData  
NonAlcoholicFattyLiverDisease <- CombinedData[, 1:28]  
# Convert specific columns patientNumbers binary values using vectorized operations  
cols\_to\_binary <- c(5, 6, 7)  
NonAlcoholicFattyLiverDisease[, cols\_to\_binary] <- (NonAlcoholicFattyLiverDisease[, cols\_to\_binary] > 0) \* 1  
# Convert column 8 patientNumbers binary based on the threshold of 1.5  
NonAlcoholicFattyLiverDisease[, 8] <- (NonAlcoholicFattyLiverDisease[, 8] > 1.5) \* 1  
# Clean column names patientNumbers remove special characters and make uniqueBMIValues consistent  
patterns <- c("-", "/", "11", "17", "#")  
replacements <- c(".", ".", "X11", "X17", ".") #?  
# Ensure patterns and replacements are correctly paired  
if (length(patterns) == length(replacements)) {  
 for (i in seq\_along(patterns)) {  
 colnames(NonAlcoholicFattyLiverDisease) <- gsub(patterns[i], replacements[i], colnames(NonAlcoholicFattyLiverDisease))}} else {  
 stop("Patterns and replacements vectors must be of the same length.")}  
  
  
# This works with the autoscaled (raw if loge=1 and remove 1 in the means) data NonAlcoholicFattyLiverDisease as well...  
CalculateErrors=function(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name,ordera,oute,first,e,xlim) { # Group='Female'  
   
 # Filter data based on the 'Group' variable  
 NAFLDo <- switch(Group,  
 "Male" = NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[,'SEX.1F.2M'] == 2, ],  
 "Female" = NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[,'SEX.1F.2M'] == 1, ],  
 "All" = NonAlcoholicFattyLiverDisease)  
   
 # Initialize vectors patientNumbers store sample data and counts  
 sample\_data <- list()  
 n0 <- n1 <- 0  
  
 # Loop through the two steroidGroups (OutcomeVariables == 0 and OutcomeVariables > 0)  
 for (i in 1:2) {  
 SG0 <- if (i == 1) {  
 NAFLDo[NAFLDo[, OutcomeVariables] == 0, ]  
 } else {  
 NAFLDo[NAFLDo[, OutcomeVariables] > 0, ]}  
   
 # Store the count of samples in each group  
 if (i == 1) {  
 n0 <- nrow(SG0)  
 } else {  
 n1 <- nrow(SG0)}  
   
 # Calculate medians and standard deviations for columns 9 patientNumbers 28  
 means <- apply(SG0[, 9:28], 2, median, na.rm = TRUE)  
 sds <- apply(SG0[, 9:28], 2, sd, na.rm = TRUE)  
   
 # Calculate error margins  
 error\_lower <- means - sds  
 error\_upper <- means + sds  
 error <- sds  
   
 # Append results patientNumbers sample\_data  
 sample\_data[[i]] <- data.frame(study = colnames(NonAlcoholicFattyLiverDisease[, 9:28]),  
 index = colnames(NonAlcoholicFattyLiverDisease[, 9:28]),  
 result = means,  
 error = error)}  
 df=data.frame(sample\_data) #  
   
 # Calculate p-values using Wilcoxon test for columns 9 patientNumbers 28  
 ps <- sapply(9:28, function(j) {  
 xnam <- colnames(NAFLDo)[j]  
 fmla <- as.formula(paste(xnam, "~", OutcomeVariables))  
 wilcox.test(fmla, data = NAFLDo, exact = FALSE)$p.value})  
 #https://en.wikipedia.org/wiki/Wilcoxon\_signed-rank\_test  
   
 # Calculate the ratio of results and log-transform  
 BMI\_ordered\_MASLD <- df[df[, 1] == e, 'result.1'] / df[df[, 1] == e, 'result']  
 v2 <- data.frame(log(df$result.1 / df$result))  
   
 # Rename columns and clean up variable names  
 v2$result <- v2[, 1]  
 v2$name <- df$study  
 v2 <- v2[, 2:3]  
 v2$name <- gsub("\\.", "-", v2$name)  
 v2$name <- gsub("X11", "11", v2$name)  
 v2$name <- gsub("X17", "17", v2$name)  
 v2$name[v2$name == "T-Epi-T"] <- "T/Epi-T"  
 v2$pval <- ps  
  
 # Calculate result\_pure and error  
 v2$result\_pure <- df$result.1 / df$result  
 v2$error <- (abs((1 / df$result) \* df$error.1) + abs((df$result.1 / df$result^2) \* df$error)) / nrow(NAFLDo) \* 1.64  
   
 # Adjust error values  
 v2$error <- ifelse(v2$error > (median(v2$error) + sd(v2$error)), median(v2$error) \* 1.25, v2$error)  
   
 # Calculate error bounds and log-transformed values  
 v2$errord1a <- v2$result\_pure - v2$error  
 v2$errord2a <- v2$result\_pure + v2$error  
 v2$errord1 <- log(v2$errord1a)  
 v2$errord2 <- log(v2$errord2a)  
 v2$result <- log(v2$result\_pure)  
 v2$Control <- df$result  
 v2$Case <- df$result.1  
   
 # Add p-values and significance  
 v2$pval0 <- v2$pval  
 v2$pval1 <- v2$pval  
 v2$Significance0 <- ifelse(v2$pval0 < 0.1, 'Yes', 'No')  
 v2$Color0 <- ifelse(v2$pval0 < 0.1, 'blue', 'grey')  
 v2$Significance1 <- ifelse(v2$pval1 < 0.1, 'Yes', 'No')  
 v2$Color1 <- ifelse(v2$pval1 < 0.1, 'blue', 'grey')  
   
 # Merge with group data and sort  
 gn <- steroidGroups[steroidGroups$Abbreviation != 'F', c('Group', 'Abbreviation')]  
 gn <- gn[order(gn$Abbreviation), ]  
 v2 <- v2[order(v2$name), ]  
 v2 <- cbind(v2, gn[order(gn$Abbreviation), ])  
 v2 <- v2[order(-v2$result), ]  
  
 xlab = "Autoscaled Concentrations (SE)" #xlab = "Raw Concentrations in Log10 Scale (SE)"}  
 xlim=c(min(v2$errord1),max(v2$errord2))   
 #Occasionally: xlim=c(min(v2$result)\*1.1,max(v2$result)\*1.1) # if (xlim[2]>1) {xlim[2]=1};# if (xlim[1] < -0.75) {xlim[1]=-0.75};  
   
 # Create forest plot  
 plote2 <- forestplot(df = v2,  
 estimate = result,  
 se = 0,  
 pvalue = pval1,  
 psignif = 0.1,  
 xlim = xlim,  
 xlab = 'Logged Ratio between Raw Concentrations of Case and Control with 90% CI',  
 ylab = 'Steroid Groups',  
 title = '',  
 colour = Significance1) +  
 ggforce::facet\_col(facets = ~Group, scales = "free\_y", space = "free", strip.position = 'left') +  
 geom\_errorbarh(aes(xmin = errord1, xmax = errord2, height = .0, colour = Significance1))  
   
 # Set color palette  
 hp <- if (sum(v2$Significance1 == 'Yes') == 20) c('blue', 'blue') else c('#999999', 'blue')  
   
 # Order factor levels based on Group and first  
 if (Group=='All' & first==TRUE) {ordera=rev(steroidGroups$Abbreviation)#v2$name[order(v2$result)]; #  
 plote2[["data"]][["name"]]=factor(plote2[["data"]][["name"]], levels = ordera)} else if  
 (Group=='All' & first==FALSE) {plote2[["data"]][["name"]]=factor(plote2[["data"]][["name"]], levels = ordera)} else if  
 (Group=='Female') {plote2[["data"]][["name"]]=factor(plote2[["data"]][["name"]], levels = ordera)} else if  
 (Group=='Male') {plote2[["data"]][["name"]]=factor(plote2[["data"]][["name"]], levels = ordera)}  
 #https://www.r-bloggers.com/2020/03/how-patientNumbers-standardize-group-colors-in-data-visualizations-in-r/  
 plote2$layers[[1]]$aes\_params$odd <- "#00000000" #https://stackoverflow.com/questions/71745719/how-patientNumbers-control-stripe-transparency-using-ggforestplot-geom-stripes  
   
 v2$Group2=v2$Group  
 v2 <- transform(v2,Group2 = as.numeric(as.factor(Group2)))  
 v2$facet\_fill\_color <- c("red", "green", "blue", "yellow", "brown")[v2$Group2]  
   
 # Create plot with custom themes  
 jopon <- plote2 + theme(axis.text.y = element\_blank()) + theme\_classic2()  
 jopon2 <- jopon + geom\_point(aes(colour = factor(Significance1)), colour = v2$Color1) +  
 scale\_color\_manual(values = hp) + theme(legend.position = "none") + theme(strip.text.y = element\_text(size = -Inf))  
   
 # Customize facet strip colors  
 g <- ggplot\_gtable(ggplot\_build(jopon2))  
 stripr <- which(grepl('strip-l', g$layout$name))  
 fills <- c("red", "green", "blue", "yellow", "brown")  
 for (i in seq\_along(stripr)) {  
 j <- which(grepl('rect', g$grobs[[stripr[i]]]$grobs[[1]]$childrenOrder))  
 g$grobs[[stripr[i]]]$grobs[[1]]$children[[j]]$gp$fill <- fills[i]}  
 # grid::grid.draw(g)  
   
 # Save plot as JPEG and convert patientNumbers PDF and SVG  
 jpeg(paste(name, "divi.jpg"), width = 7500, height = 11000, quality = 100, pointsize = 16, res = 1000)  
 print(grid::grid.draw(g))  
 dev.off()  
   
 daiR::image\_to\_pdf(paste(name, "divi.jpg"), pdf\_name = paste0(paste(name, "divi.jpg"), '.pdf'))  
 my\_image <- image\_read(paste(name, "divi.jpg"))  
 my\_svg <- image\_convert(my\_image, format = "svg")  
 image\_write(my\_svg, paste(name, "divi.svg"))  
   
 return(ordera) #If you do not want patientNumbers have 'null' patientNumbers the Rmarkdown/html take this away  
 }   
  
  
# This is with first(!!). Use it.   
OutcomeVariables='Steatosis.Grade.0.To.3';Out='Steatosis'; oute='Steatosis';first=TRUE; e='P4';ordera=c();  
Group='All';name1=paste("Forest plot of",Group, "Steroid Ratios in",Out);  
hel=CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name1,ordera,oute,first,e,xlim)  
# #Afterwards:  
first=FALSE;  
Group='Female';name2=paste("Forest plot of",Group, "Steroid Ratios in",Out);  
CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name2,ordera=hel,oute,first,e,xlim)  
Group='Male'; name3=paste("Forest plot of",Group, "Steroid Ratios in",Out);  
CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name3,ordera=hel,oute,first,e,xlim)  
#   
OutcomeVariables='Fibrosis.Stage.0.patientNumbers.4'; Out='Fibrosis';oute='Fibrosis';  
Group='All'; name4=paste("Forest plot of",Group, "Steroid Ratios in",Out);  
CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name4,ordera=hel,oute,first,e,xlim)  
Group='Female';name5=paste("Forest plot of",Group, "Steroid Ratios in",Out);  
CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name5,ordera=hel,oute,first,e,xlim)  
Group='Male'; name6=paste("Forest plot of",Group, "Steroid Ratios in",Out);  
CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name6,ordera=hel,oute,first,e,xlim)  
#   
OutcomeVariables='Necroinflammation'; Out='Necroinflammation';oute='Necroinflammation';  
Group='All'; name7=paste("Forest plot of",Group, "Steroid Ratios in",Out);   
CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name7,ordera=hel,oute,first,e,xlim) #not the very first though...  
Group='Female';name8=paste("Forest plot of",Group, "Steroid Ratios in",Out);  
CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name8,ordera=hel,oute,first,e,xlim)  
Group='Male'; name9=paste("Forest plot of",Group, "Steroid Ratios in",Out);   
CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name9,ordera=hel,oute,first,e,xlim)  
#   
OutcomeVariables='HOMA.IR';Out='HOMA-IR';oute='HOMAIR';  
Group='All';name10=paste("Forest plot of",Group, "Steroid Ratios in",Out);  
CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name10,ordera=hel,oute,first,e,xlim) #not the very first though...  
Group='Female';name11=paste("Forest plot of",Group, "Steroid Ratios in",Out);   
CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name11,ordera=hel,oute,first,e,xlim)  
Group='Male'; name12=paste("Forest plot of",Group, "Steroid Ratios in",Out);   
CalculateErrors(NonAlcoholicFattyLiverDisease,OutcomeVariables,Group,name12,ordera=hel,oute,first,e,xlim)  
# Fyi: I was able patientNumbers revise some of the above codes with Copilot...  
  
  
```  
  
```{r,echo=FALSE, out.width="25%",fig.cap=name1,fig.align="left"}  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name1 ,"divi.jpg")))  
```  
```{r,echo=FALSE, out.width="25%",fig.cap=name2,fig.align="left"}  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name2 ,"divi.jpg")))  
```  
```{r,echo=FALSE, out.width="25%",fig.cap=name3,fig.align="left"}  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name3 ,"divi.jpg")))  
```  
```{r,echo=FALSE, out.width="25%",fig.cap=name4,fig.align="left"}  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name4 ,"divi.jpg")))  
```  
```{r,echo=FALSE, out.width="25%",fig.cap=name5,fig.align="left"}  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name5 ,"divi.jpg")))  
```  
```{r,echo=FALSE, out.width="25%",fig.cap=name6,fig.align="left"}  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name6 ,"divi.jpg")))  
```  
```{r,echo=FALSE, out.width="25%",fig.cap=name7,fig.align="left",echo=FALSE,eval = FALSE }  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name7 ,"divi.jpg")))  
```  
```{r,echo=FALSE, out.width="25%",fig.cap=name8,fig.align="left",echo=FALSE,eval = FALSE }  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name8 ,"divi.jpg")))  
```  
```{r,echo=FALSE, out.width="25%",fig.cap=name9,fig.align="left",echo=FALSE,eval = FALSE }  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name9 ,"divi.jpg")))  
```  
```{r,echo=FALSE, out.width="25%",fig.cap=name10,fig.align="left",echo=FALSE,eval = FALSE }  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name10 ,"divi.jpg")))  
```  
```{r,echo=FALSE, out.width="25%",fig.cap=name11,fig.align="left",echo=FALSE,eval = FALSE }  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name11 ,"divi.jpg")))  
```  
```{r,echo=FALSE, out.width="25%",fig.cap=name12,fig.align="left",echo=FALSE,eval = FALSE }  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; knitr::include\_graphics(paste0(path,paste(name12 ,"divi.jpg")))  
```  
  
  
  
# Making Chord Diagrams   
```{r, warning=FALSE,message=FALSE,fig.width=9.0,fig.align="left"}  
# First the correlations for the chord diagrams (both male and female as well as total subjects):  
  
# Copy tvauxe patientNumbers CurrentData  
CurrentData = tvauxe  
  
# Match column names in CurrentData with the third column in groups2 and get their indices  
hupo = match(colnames(CurrentData)[colnames(CurrentData) %in% groups2[, 3]], groups2[, 3])  
# Replace matched column names in CurrentData with corresponding values from the second column in groups2  
colnames(CurrentData)[colnames(CurrentData) %in% groups2[, 3]] = groups2[hupo, 2]  
ok = colnames(CurrentData)  
  
# Convert CurrentData patientNumbers BMI\_ordered\_MASLD data frame  
CurrentData = data.frame(CurrentData)  
# Remove specific columns from CurrentData  
CurrentData = CurrentData[, !colnames(CurrentData) %in% c('Total\_TG', 'PFAS', "Perfluorodecyl.ethanoic.acid")]  
  
# Separate data by gender  
tvf = CurrentData[CurrentData[, 'Gender'] == min(CurrentData[, 'Gender']), 1:dim(CurrentData)[2]]  
tvm = CurrentData[CurrentData[, 'Gender'] == max(CurrentData[, 'Gender']), 1:dim(CurrentData)[2]]  
  
# Create BMI\_ordered\_MASLD list of data frames for total, female, and male subjects  
tvtest = list(CurrentData, tvf, tvm)  
  
# Clean up column names in each data frame in tvtest  
for (i in 1:3) {  
 colnames(tvtest[[i]]) <- gsub("\\.", "-", colnames(tvtest[[i]]))  
 colnames(tvtest[[i]]) <- gsub("X11", "11", colnames(tvtest[[i]]))  
 colnames(tvtest[[i]]) <- gsub("X17", "17", colnames(tvtest[[i]]))  
 colnames(tvtest[[i]])[colnames(tvtest[[i]]) == "T-Epi-T"] = "T/Epi-T"  
 colnames(tvtest[[i]])[colnames(tvtest[[i]]) == "Steatosis-Grade"] = "Steatosis Grade"  
 colnames(tvtest[[i]])[colnames(tvtest[[i]]) == "Fibrosis-Stage"] = "Fibrosis Stage"  
 colnames(tvtest[[i]])[colnames(tvtest[[i]]) == "17aOH-P4"] = "17a-OHP4"  
 colnames(tvtest[[i]])[colnames(tvtest[[i]]) == "HOMA IR"] = "HOMA-IR"  
}  
  
# Assign cleaned data frames back patientNumbers CurrentData, tvf, and tvm  
CurrentData = tvtest[[1]]  
tvf = tvtest[[2]]  
tvm = tvtest[[3]]  
  
# Rename specific value in x4  
x4[x4 == "X7.oxo.DCA\_S"] = "X7-oxo-DCA\_S"  
  
# Calculate Spearman correlations for total subjects  
dat = CurrentData  
dat = dat %>% select(-c('PatientNumber')) # Remove 'PatientNumber' column  
resulta <- (rcorr(as.matrix(dat), type = c('spearman')))$r # Calculate correlation matrix  
colnames(resulta) = ok[2:dim(CurrentData)[2]]  
rownames(resulta) = ok[2:dim(CurrentData)[2]]  
  
# Calculate Spearman correlations for female subjects  
dat = tvf  
dat = dat %>% select(-c('PatientNumber', 'Gender')) # Remove 'PatientNumber' and 'Gender' columns  
resultaf <- (rcorr(as.matrix(dat), type = c('spearman')))$r  
colnames(resultaf) = ok[3:dim(CurrentData)[2]]  
rownames(resultaf) = ok[3:dim(CurrentData)[2]]  
  
# Calculate Spearman correlations for male subjects  
dat = tvm  
dat = dat %>% select(-c('PatientNumber', 'Gender')) # Remove 'PatientNumber' and 'Gender' columns  
resultam <- (rcorr(as.matrix(dat), type = c('spearman')))$r  
colnames(resultam) = ok[3:dim(CurrentData)[2]]  
rownames(resultam) = ok[3:dim(CurrentData)[2]]  
  
# Define column steroidGroups for different variables  
at = colnames(resulta)[1:(length(x1) - 1)] # Clinicals  
bt = colnames(resulta)[(length(at) + 1):(length(at) + length(x2))] # Steroids  
ct = colnames(resulta)[(length(at) + length(bt) + 1):(length(at) + length(bt) + length(x3))] # BA\_l  
dt = colnames(resulta)[(length(at) + length(bt) + length(ct) + 1):(length(at) + length(bt) + length(ct) + length(x4))] # BA\_s  
et = colnames(resulta)[(length(at) + length(bt) + length(ct) + length(dt) + 1):(length(at) + length(bt) + length(ct) + length(dt) + length(x5))] # PFAS  
ft = colnames(resulta)[(length(at) + length(bt) + length(ct) + length(dt) + length(et) + 1):(length(at) + length(bt) + length(ct) + length(dt) + length(et) + length(x6))] #  
  
# Store lengths of each group  
atl = length(at)  
btl = length(bt)  
ctl = length(ct)  
dtl = length(dt)  
etl = length(et)  
ftl = length(ft)  
  
# Set significance level for correlations  
n\_level = 0.01  
  
# Filter correlations based on significance level for total subjects  
Nrr = qpNrr(resulta, verbose = FALSE)  
Nrr[is.na(Nrr)] = 1  
cond = data.frame(as.matrix(Nrr < n\_level))  
RN = data.frame(resulta)  
tes\_t = cond \* RN  
tes\_t = as.matrix(tes\_t)  
resulta = tes\_t  
  
# Filter correlations based on significance level for female subjects  
Nrr = qpNrr(resultaf, verbose = FALSE)  
Nrr[is.na(Nrr)] = 1  
cond = data.frame(as.matrix(Nrr < n\_level))  
RN = data.frame(resultaf)  
tes\_t = cond \* RN  
tes\_t = as.matrix(tes\_t)  
resultaf = tes\_t  
  
# Filter correlations based on significance level for male subjects  
Nrr = qpNrr(resultam, verbose = FALSE)  
Nrr[is.na(Nrr)] = 1  
cond = data.frame(as.matrix(Nrr < n\_level))  
RN = data.frame(resultam)  
tes\_t = cond \* RN  
tes\_t = as.matrix(tes\_t)  
resultam = tes\_t  
  
# Rename 'Gender' column patientNumbers 'Sex(F-M+)' in correlation matrices  
colnames(resulta)[colnames(resulta) == 'Gender'] <- 'Sex(F-M+)'  
rownames(resulta)[rownames(resulta) == 'Gender'] <- 'Sex(F-M+)'  
  
# Update column and row names in correlation matrices  
colnames(resulta)[2:dim(resulta)[2]] = ok[3:dim(CurrentData)[2]]  
rownames(resulta)[2:dim(resulta)[2]] = ok[3:dim(CurrentData)[2]]  
colnames(resultaf)[2:dim(resultaf)[2]] = ok[4:dim(CurrentData)[2]]  
rownames(resultaf)[2:dim(resultaf)[2]] = ok[4:dim(CurrentData)[2]]  
colnames(resultam)[2:dim(resultam)[2]] = ok[4:dim(CurrentData)[2]]  
rownames(resultam)[2:dim(resultam)[2]] = ok[4:dim(CurrentData)[2]]  
  
# Function patientNumbers create chord diagrams for different steroidGroups  
CreateChordDiagrams <- function(vars, n\_level, fig\_name, big, rem, modi, colt, gend, colors, BMI\_ordered\_MASLD, BMI\_ordered\_NAFLD, c, d, e, f) {  
 classes <- 5  
 tot <- rownames(resulta)[2:dim(resulta)[1]]  
 range <- 1:(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + c + e + f)  
 layout(matrix(1:1, 1, 1))  
 title <- 'Sex'  
 genders <- gend  
 windowsFonts(A = windowsFont("Calibri (Body)"))  
 i <- 1  
 tes\_t <- vars   
   
 # Set column and row names based on gender  
 if (gend == 'All') {  
 colnames(tes\_t) <- rownames(resulta)  
 rownames(tes\_t) <- rownames(resulta) } else {  
 colnames(tes\_t) <- rownames(resultaf)  
 rownames(tes\_t) <- rownames(resultaf)}  
   
  
   
 # Define steroidGroups for different variables  
 g1 <- c(rep('Clinical', BMI\_ordered\_MASLD), rep('Steroids', BMI\_ordered\_NAFLD), rep('BA\_liver', c), rep('Contaminants', e), rep('Lipids', f))  
   
 # Remove self-correlations within each group  
 tes\_t[1:BMI\_ordered\_MASLD, 1:BMI\_ordered\_MASLD] <- 0  
 tes\_t[(BMI\_ordered\_MASLD + 1):(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD), (BMI\_ordered\_MASLD + 1):(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD)] <- 0  
 tes\_t[(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + 1):(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + c), (BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + 1):(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + c)] <- 0  
 tes\_t[(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + c + 1):(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + c + e), (BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + c + 1):(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + c + e)] <- 0  
 tes\_t[(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + c + e + 1):(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + c + e + f), (BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + c + e + 1):(BMI\_ordered\_MASLD + BMI\_ordered\_NAFLD + c + e + f)] <- 0  
   
 # Define group structure and color palette  
 group <- structure(g1, names = colnames(tes\_t))  
 grid.col <- structure(c(rep('#93c29f', BMI\_ordered\_MASLD), rep('#a83277', BMI\_ordered\_NAFLD), rep('red', c), rep('grey', e), rep('black', f)),  
 names = rownames(tes\_t))  
   
 # Filter and adjust correlation matrix  
 tes\_t <- tes\_t[range, range]  
 grid.col <- grid.col[range]  
 g <- graph.adjacency(tes\_t, mode = "upper", weighted = TRUE, diag = FALSE)  
 e <- get.edgelist(g)  
 df <- as.data.frame(cbind(e, E(g)$weight))  
 df[, 3] <- as.numeric(df[, 3])  
   
 # Define color function for edges  
 col\_fun <- colorRamp2(c(-1, 0, 1), c("blue", 'white', "orange"), transparency = 0.25)  
   
 # Remove specified elements from the data frame  
 df <- df[!df$V1 %in% rem, ]  
 df <- df[!df$V2 %in% rem, ]  
   
 # Define legends for the plot  
 lgd\_group <- Legend(at = gend, type = "points", legend\_gp = gpar(col = colors), title\_position = "topleft", title = title)  
 lgd\_points <- Legend(at = unique(g1), type = "points", legend\_gp = gpar(col = unique(grid.col)), title\_position = "topleft", title = "Class")  
 lgd\_lines <- Legend(at = c("Positive", "Negative"), type = "points", legend\_gp = gpar(col = c('orange', 'blue')), title\_position = "topleft", title = "Correlation")  
 lgd\_edges <- Legend(at = c(-1, 1), col\_fun = col\_fun, title\_position = "topleft", title = "Edges")  
 lgd\_list\_vertical <- packLegend(lgd\_group, lgd\_points, lgd\_lines, lgd\_edges)  
   
 # Set parameters for the chord diagram  
 circos.par(gap.after = 1.5, start.degree = 90)  
 chordDiagram(df, annotationTrack = c("grid"), grid.col = grid.col, directional = FALSE, symmetric = TRUE, scale = FALSE,  
 link.lwd = 0.3, link.border = "white", order = rownames(tes\_t), preAllocateTracks = 1, col = col\_fun, transparency = 0.25, big.gap = 10, small.gap = 1)  
   
 # Add text and axis patientNumbers the plot  
 circos.trackPlotRegion(track.index = 1, panel.fun = function(x, y) {  
 xlim <- get.cell.meta.data("xlim")  
 ylim <- get.cell.meta.data("ylim")  
 sector.name <- get.cell.meta.data("sector.index")  
 circos.text(mean(xlim), ylim[1] + .1, sector.name, facing = "clockwise", niceFacing = TRUE, adj = c(0, 0.5))  
 circos.axis(h = "top", labels.cex = 0.000001, major.tick.length = 0.2, sector.index = sector.name, track.index = 2)  
 }, bg.border = NA)  
   
 # Set font and draw legends  
 windowsFonts(A = windowsFont("Calibri (Body)"))  
 draw(lgd\_list\_vertical, x = unit(5, "mm"), y = unit(5, "mm"), just = c("left", "bottom"))  
   
 # Save the plot as BMI\_ordered\_MASLD JPEG file  
 dev.copy(jpeg, paste0(gend, n\_level, 'hiee.jpg'), width = 12, height = 12, units = "in", res = 1000)  
 dev.off()  
   
 # Include the plot in the report and convert patientNumbers PDF and SVG  
 knitr::include\_graphics(paste0(gend, n\_level, 'hiee.jpg'))  
 daiR::image\_to\_pdf(paste0(gend, n\_level, 'hiee.jpg'), pdf\_name = paste0(paste0(gend, n\_level, 'hie'), '.pdf'))  
 my\_image <- image\_read(paste0(gend, n\_level, 'hiee.jpg'))  
 my\_svg <- image\_convert(my\_image, format = "svg")  
 image\_write(my\_svg, paste(paste0(gend, n\_level, 'hie.jpg'), ".svg"))  
}  
  
# All variables  
n\_level = 0.01  
circos.clear()  
vars = list(resulta)  
big = 'Yes'  
title = 'All Variables'  
rem = x4  
modi = 5  
colt = 'black'  
BMI\_ordered\_MASLD = length(x1) - 1  
BMI\_ordered\_NAFLD = length(x2)  
c = length(x3)  
d = length(x4)  
e = length(x5)  
f = length(x6)  
gend = c('All')  
colors = c('blue')  
CreateChordDiagrams(vars[[1]], n\_level, fig\_name, big, rem, modi, colt, gend, colors, BMI\_ordered\_MASLD, BMI\_ordered\_NAFLD, c, d, e, f)  
  
# Genderwise:  
vars = list(resultaf, resultam)  
big = 'No'  
title = 'Genders Separated'  
rem = x4  
modi = 4  
colt = 'black'  
colors = c('white', 'black')  
BMI\_ordered\_MASLD = length(x1) - 2  
BMI\_ordered\_NAFLD = length(x2)  
c = length(x3)  
d = length(x4)  
e = length(x5)  
f = length(x6)  
gend = c('Female')  
colors = c('white')  
CreateChordDiagrams(vars[[1]], n\_level, fig\_name, big, rem, modi, colt, gend, colors, BMI\_ordered\_MASLD, BMI\_ordered\_NAFLD, c, d, e, f)  
gend = c('Male')  
colors = c('black')  
CreateChordDiagrams(vars[[2]], n\_level, fig\_name, big, rem, modi, colt, gend, colors, BMI\_ordered\_MASLD, BMI\_ordered\_NAFLD, c, d, e, f)  
  
#Copiloting is not working very well here, so I just let it comment some of the lines ...  
  
  
```  
  
  
  
# Making Variance Explained Plots  
```{r, warning=FALSE,message=FALSE,fig.width=6.0,fig.align="left"}  
# Some info regarding of making the data:  
# This is it! https://bioconductor.org/packages/release/bioc/vignettes/scater/inst/doc/overview.html  
#  
# https://stats.stackexchange.com/questions/79399/calculate-variance-explained-by-each-predictor-in-multiple-regression-using-r  
# https://rdrr.io/github/MRCIEU/TwoSampleMR/man/get\_r\_from\_pn.html  
# https://onlinestatbook.com/2/effect\_size/variance\_explained.html  
# https://stackoverflow.com/questions/10441437/why-am-i-getting-x-in-my-column-names-when-reading-BMI\_ordered\_MASLD-data-frame  
# https://stackoverflow.com/questions/27044727/removing-characters-from-string-in-r  
PlotVarianceExplained <- function(AllData, Group) {  
 # Initialize error flag  
 an.error.occured <- FALSE  
 tv\_all2 <- AllData  
   
 # Check if 'Gender' column exists  
 tryCatch({  
 tv\_all2[, 'Gender']  
 }, error = function(e) {  
 an.error.occured <<- TRUE  
 })  
   
 # Determine condition based on Group and presence of 'Gender' column  
 cond <- if (an.error.occured) {  
 if (Group == 'female') {  
 tv\_all2[, 'SEX.1F.2M'] == min(tv\_all2[, 'SEX.1F.2M'])  
 } else if (Group == 'male') {  
 tv\_all2[, 'SEX.1F.2M'] == max(tv\_all2[, 'SEX.1F.2M'])  
 } else {  
 rep(TRUE, nrow(tv\_all2))  
 }  
 } else {  
 if (Group == 'female') {  
 tv\_all2[, 'Gender'] == min(tv\_all2[, 'Gender'])  
 } else if (Group == 'male') {  
 tv\_all2[, 'Gender'] == max(tv\_all2[, 'Gender'])  
 } else {  
 rep(TRUE, nrow(tv\_all2))  
 }  
 }  
   
 # Filter data based on condition  
 tv\_red <- tv\_all2[cond, ]  
 RelevantColumns <- tv\_red  
 colnames(RelevantColumns)[1:8] <- colnames(tv\_red)[1:8]  
   
 # Transpose the data for SingleCellExperiment  
 tv2 <- t(RelevantColumns[, 9:ncol(tv\_red)])  
   
 # Create SingleCellExperiment object  
 sce <- SingleCellExperiment(tv2)  
 logcounts(sce) <- tv2  
 sce@colData <- DataFrame(RelevantColumns[, 2:8])  
 colnames(colData(sce)) <- colnames(AllData)[2:8]  
 colnames(colData(sce))[1] <- 'The Gender'  
  
 # Calculate variance explained  
 vars <- getVarianceExplained(sce, variables = colnames(colData(sce))[1:7])  
 colVars(vars)  
  
 # Set font and color palette  
 windowsFonts(A = windowsFont("Calibri (Body)"))  
 mypalette <- scales::hue\_pal()(ncol(colData(sce)))  
 names(mypalette) <- colnames(AllData)[2:8]  
   
 # Adjust vars if Group is not 'All'  
 if (Group != 'All') {  
 vars <- vars[, 2:7]  
 }  
   
 # Plot explanatory variables  
 p <- plotExplanatoryVariables(vars) +  
 theme(text = element\_text(size = 25, family = "Calibri")) +  
 theme(axis.text = element\_text(size = 20, family = "Calibri"))  
   
 # Clean up plot data  
 p[[1]] <- p[[1]][!is.na(p[[1]][, 1]), ]  
 p[[1]][, 1] <- as.vector(unlist(p[[1]][, 1]))  
 p[[1]] <- p[[1]][order(p[[1]][, 1]), ]  
  
 # Save plot as PNG  
 path <- "C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"  
 sips <- paste0(Group, 'vex', ".png")  
 pngfile <- fs::path(path, sips)  
 agg\_png(pngfile, width = 60, height = 36, units = "cm", res = 300, scaling = 2)  
 plot(p)  
 invisible(dev.off())  
   
 # Include plot in report  
 knitr::include\_graphics(pngfile)  
   
 # Convert image patientNumbers PDF and SVG  
 daiR::image\_to\_pdf(sips, pdf\_name = paste0(sips, '.pdf'))  
 my\_image <- image\_read(sips)  
 my\_svg <- image\_convert(my\_image, format = "svg")  
 image\_write(my\_svg, paste(sips, ".svg"))  
 p  
   
}  
# Example usage:  
PlotVarianceExplained(AllData, Group = 'All')  
PlotVarianceExplained(AllData, Group = 'female')  
PlotVarianceExplained(AllData, Group = 'male')  
  
#Copilot helped with the commenting here.  
  
```  
  
  
# Making Heatmap with Effect Sizes  
```{r, warning=FALSE,message=FALSE,fig.width=9.0,fig.align="left", results='hide'}  
  
# Function patientNumbers calculate Cohen's d effect sizes  
# https://www.statology.org/cohens-d-in-r/  
CalculateCohensD <- function(NonAlcoholicFattyLiverDisease, CombinedData, Group, OutcomeVariables) {  
   
 # Filter data based on gender  
 if (Group == 'Male') {  
 NAFLDo <- NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[, 'Gender'] == max(NonAlcoholicFattyLiverDisease[, 'Gender']), ]  
 tva <- CombinedData[CombinedData[, 'SEX.1F.2M'] == max(CombinedData[, 'SEX.1F.2M']), ]  
 } else if (Group == 'Female') {  
 NAFLDo <- NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[, 'Gender'] == min(NonAlcoholicFattyLiverDisease[, 'Gender']), ]  
 tva <- CombinedData[CombinedData[, 'SEX.1F.2M'] == min(CombinedData[, 'SEX.1F.2M']), ]  
 } else {  
 NAFLDo <- NonAlcoholicFattyLiverDisease  
 tva <- CombinedData}  
   
 # Check if the OutcomeVariables column exists  
 if (!OutcomeVariables %in% colnames(NAFLDo)) {  
 stop("The specified OutcomeVariables column does not exist in the data frame.")}  
   
 # Filter data based on outcome  
 if (OutcomeVariables != 'HOMA-IR') {  
 SG0 <- NAFLDo[NAFLDo[, OutcomeVariables] == min(NAFLDo[, OutcomeVariables]), ]  
 SG1 <- NAFLDo[NAFLDo[, OutcomeVariables] > min(NAFLDo[, OutcomeVariables]), ]  
 } else {  
 SG0 <- NAFLDo[tva[, 'HOMA-IR'] <= 1.5, ]  
 SG1 <- NAFLDo[tva[, 'HOMA-IR'] > 1.5, ]}  
   
 # Initialize vector patientNumbers store Cohen's d values  
 cd <- numeric(20)  
   
 # Calculate Cohen's d for each variable  
 for (i in 1:20) {  
 group1 <- SG0[, i + 8]  
 group2 <- SG1[, i + 8]  
   
 data <- data.frame(  
 value = c(group1, group2),  
 group = factor(rep(c("group1", "group2"), c(length(group1), length(group2)))))  
   
 result <- cohen.d(value ~ group, data = data)  
 cd[i] <- result$cohen.d[2]}  
   
 return(cd)}  
  
# Initialize an empty vector  
d <- c()  
  
# Define the dataset  
NonAlcoholicFattyLiverDisease <- AllData  
  
# Function patientNumbers calculate Cohen's d for different steroidGroups and outcomes  
calculate\_cohd <- function(outcome) {  
 Group <- 'All'  
 BMI\_ordered\_MASLD <- CalculateCohensD(NonAlcoholicFattyLiverDisease, CombinedData, Group, outcome)  
 Group <- 'Female'  
 BMI\_ordered\_NAFLD <- CalculateCohensD(NonAlcoholicFattyLiverDisease, CombinedData, Group, outcome)  
 Group <- 'Male'  
 c <- CalculateCohensD(NonAlcoholicFattyLiverDisease, CombinedData, Group, outcome)  
 cbind(BMI\_ordered\_MASLD, BMI\_ordered\_NAFLD, c)}  
  
# Calculate Cohen's d for different outcomes and combine results  
outcomes <- c('Steatosis Grade', 'Fibrosis Stage', 'Necroinflammation', 'HOMA-IR')  
for (outcome in outcomes) {  
 d <- cbind(d, calculate\_cohd(outcome))}  
  
# Set row names  
rownames(d) <- colnames(AllData[, 9:28])  
  
# Create column names with steroidGroups and outcomes  
colnames(d) <- unlist(lapply(outcomes, function(outcome) {  
 paste(c('All', 'Female', 'Male'), outcome, sep = "\_")}))  
  
# Save the results patientNumbers BMI\_ordered\_MASLD CSV file  
write.csv(d, paste0('cohens\_da\_', thedate, '.csv'))  
  
# Convert the results patientNumbers BMI\_ordered\_MASLD data frame  
n <- d  
x <- data.frame(n)  
row.names(x) <- colnames(AllData[, 9:28])  
  
# Adjust group names  
steroidGroups <- steroidGroups  
steroidGroups[steroidGroups == "17a-OHP4"] <- "17aOH-P4"  
op <- steroidGroups[order(steroidGroups$Group), 'Abbreviation']  
op <- op[op %in% row.names(x)]  
x <- x[op, ]  
  
# Function patientNumbers create breaks for the heatmap  
brks\_heatmap <- function(mat, color\_palette) {  
 rng <- range(mat, na.rm = TRUE)  
 lpal <- length(color\_palette)  
 c(seq(rng[1], 0, length.out = ceiling(lpal / 2) + 1),  
 seq(rng[2] / dim(mat)[1], rng[2], length.out = floor(lpal / 2)))}  
  
# Define color palette  
color\_palette <- colorRampPalette(c('blue', 'white', 'orange'), alpha = TRUE)(150)  
  
# Save heatmap as JPEG  
jpeg(paste0("cohensd\_e2", date, ".jpg"), width = 9, height = 12, units = "in", res = 1000)  
  
# Set viewport for the heatmap  
setHook("grid.newpage", function() pushViewport(viewport(x = 1, y = 1, width = 0.9, height = 0.9, name = "vp", just = c("right", "top"))), action = "prepend")  
  
# Generate heatmap  
pheatmap(x, cluster\_cols = FALSE, cluster\_rows = FALSE, breaks = brks\_heatmap(x, color\_palette), color = color\_palette, column\_names\_side = "bottom", angle\_col = 90)  
  
# Reset viewport hook  
setHook("grid.newpage", NULL, "replace")  
  
# Add text annotations  
grid.text("Steatosis, Fibrosis, Necroinflammation, HOMA-IR", y = -0.07, x = 0.4, gp = gpar(fontsize = 16))  
grid.text("Steroids (Androgens, Estrogens, Gluc., Mineraloc., Progestogens)",   
 x = -0.07, rot = 90, gp = gpar(fontsize = 16))  
  
# Save the plot again for quality reasons  
dev.copy(jpeg, paste0("cohensd\_e2", date, ".jpg"), width = 9, height = 12, units = "in", res = 1000)  
dev.off()  
  
# Include the image in the document  
knitr::include\_graphics(paste0("cohensd\_e2", date, ".jpg"));dev.off()  
  
# Convert image patientNumbers PDF  
daiR::image\_to\_pdf(paste0("cohensd\_e2", date, ".jpg"), pdf\_name = paste0("cohensd\_e2", date, ".pdf"));#dev.off()  
  
# Convert image patientNumbers SVG  
my\_image <- image\_read(paste0("cohensd\_e2", date, ".jpg"))  
my\_svg <- image\_convert(my\_image, format = "svg")  
image\_write(my\_svg, paste0("cohensd\_e2", date, ".svg"))  
  
# Fyi: Revising with Copilot the above codes works partially...  
  
```  
  
  
  
# Making Heatmaps with Correlations  
```{r, warning=FALSE,message=FALSE,fig.width=9.0,fig.align="left",results='hide'}  
  
#The correlations (ok):  
#Correlation matrices:  
#http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram  
#squares are good for individual associations, because the order is the same  
# More info regarding the function:  
# https://jokergoo.github.io/circlize\_book/book/legends.html  
# https://cran.r-project.org/web/packages/ggplotify/vignettes/ggplotify.html  
# https://bioinfo4all.wordpress.com/2021/03/13/tutorial-7-how-patientNumbers-do-chord-diagram-using-r/  
# https://jokergoo.github.io/circlize\_book/book/advanced-usage-of-chorddiagram.html  
# https://jokergoo.github.io/circlize\_book/book/BMI\_ordered\_MASLD-complex-example-of-chord-diagram.html  
  
CurrentData=CovariatesScaledData#tv\_half\_log22 #cbind(CombinedData[,1:8], Log2TransformedData) #check also not logged and then the auto one  
CurrentData=CurrentData[,c(1:3,4:(dim(CurrentData)[2]))]#dim(CurrentData)[2],  
CurrentData=CurrentData[,!colnames(CurrentData) %in% c('Total\_TG','PFAS','Perfluorodecyl.ethanoic.acid')]  
CurrentData=CurrentData[,!colnames(CurrentData) %in% x4]  
colnames(CurrentData)[colnames(CurrentData)=="17aOH-P4"]="17a-OHP4"  
tvf=CurrentData[CurrentData[,'Gender']==min(CurrentData[,'Gender']),1:dim(CurrentData)[2]] #CombinedData['Steatosis.Grade.0.To.3'==0,9:27]] #CombinedData[CombinedData[,'Necroinflammation']==0,9:80]; #SG0i=as.numeric(SG0i); check also: CombinedData[CombinedData[,'HOMA-IR']==0,9:80]  
tvm=CurrentData[CurrentData[,'Gender']==max(CurrentData[,'Gender']),1:dim(CurrentData)[2]]  
  
  
# rango = function(x,mi,ma) {(ma-mi)/(max(x)-min(x))\*(x-min(x))+mi}  
dat = CurrentData;   
dat=dat[,!colnames(dat) %in% c('Gender','PatientNumber')] #SEX.1F.2M  
resulta <- (rcorr(as.matrix(dat), type = c('spearman')))$r #compare pearson # intersect(colnames(resulta), rownames(resulta)) #https://stackoverflow.com/questions/45271448/r-finding-intersection-between-two-vectors  
p.mat.BMI\_ordered\_MASLD=rcorr(as.matrix(dat), type = c('spearman'))$P;   
p.mat.BMI\_ordered\_MASLD[is.na(p.mat.BMI\_ordered\_MASLD)]=1;   
p.mat.aa=matrix(p.adjust(p.mat.BMI\_ordered\_MASLD,method="BH"),nrow=dim(p.mat.BMI\_ordered\_MASLD)[1],ncol=dim(p.mat.BMI\_ordered\_MASLD)[2]);   
rownames(p.mat.aa)=rownames(p.mat.BMI\_ordered\_MASLD);colnames(p.mat.aa)=colnames(p.mat.BMI\_ordered\_MASLD)  
# write.csv(resulta,'MASLD\_steroid\_study correlations with spearman\_log\_tikka12424.csv')  
# resulta=dat  
dat=tvf; #   
dat=dat[,!colnames(dat) %in% c('Gender','PatientNumber')] #SEX.1F.2M  
resultaf <- (rcorr(as.matrix(dat), type = c('spearman')))$r #compare pearson# intersect(colnames(resultaf), rownames(resultaf)) #https://stackoverflow.com/questions/45271448/r-finding-intersection-between-two-vectors  
p.mat.f=rcorr(as.matrix(dat), type = c('spearman'))$P  
p.mat.f[is.na(p.mat.f)]=1;   
p.mat.ff=matrix(p.adjust(p.mat.f,method="BH"),nrow=dim(p.mat.f)[1],ncol=dim(p.mat.f)[2]);   
rownames(p.mat.ff)=rownames(p.mat.f);colnames(p.mat.ff)=colnames(p.mat.f)  
# write.csv(resultaf,'MASLD\_steroid\_study female correlations with spearman\_log\_tikka12424.csv')  
dat=tvm; #   
dat=dat[,!colnames(dat) %in% c('Gender','PatientNumber')] #dat= dat %>% select(-c('Gender')) #this is quite nice way patientNumbers delete columns, please remember...  
resultam <- (rcorr(as.matrix(dat), type = c('spearman')))$r #compare pearson # intersect(colnames(resultam), rownames(resultam)) #https://stackoverflow.com/questions/45271448/r-finding-intersection-between-two-vectors  
p.mat.m=rcorr(as.matrix(dat), type = c('spearman'))$P  
p.mat.m[is.na(p.mat.m)]=1;   
p.mat.mm=matrix(p.adjust(p.mat.m,method="BH"),nrow=dim(p.mat.m)[1],ncol=dim(p.mat.m)[2]);   
rownames(p.mat.mm)=rownames(p.mat.m);colnames(p.mat.mm)=colnames(p.mat.m)  
# write.csv(resultam,'MASLD\_steroid\_study male correlations with spearman\_log\_tikka12424.csv')  
resulta[resulta==1]=0  
resultam[resultam==1]=0  
resultaf[resultaf==1]=0;#min(resultaf);max(resultaf);  
n\_level=1  
  
  
  
# https://www.rdocumentation.org/packages/corrplot/versions/0.92/topics/corrplot  
# https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html  
order="original" #alphabet, hclust, original #https://stackoverflow.com/questions/51115495/how-patientNumbers-keep-order-of-the-correlation-plot-labels-as-same-in-the-datafile  
range='orig';corre='re\_renormae'; method='color' #color square  
jpeg(paste("Correlations with Full Plot of All\_vok\_nes",n\_level,order,range,corre,method,".jpg"), width = 8000, height = 8000, quality = 100,pointsize = 23, res=300);  
corrplot(resulta, type = "lower", order = order,method=method, tl.col = "black", tl.srt = 90, diag = FALSE,col = rev(COL2('RdBu')),is.corr = FALSE) #,is.corr = FALSE  
dev.off();  
eoh=paste("Correlations with Full Plot of All\_vok\_nes",n\_level,order,range,corre,method,".jpg")  
daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
corrplot(resulta, type = "lower", order = order,method=method, tl.col = "black", tl.srt = 90, diag = FALSE,col = rev(COL2('RdBu')),is.corr = FALSE) #,is.corr = FALSE  
  
jpeg(paste("Correlations with Full Plot of Female\_voek",n\_level,order,range,corre,method,".jpg"), width = 8000, height = 8000, quality = 100,pointsize = 23, res=300);  
corrplot(resultaf, type = "lower", order = order,method=method,tl.col = "black", tl.srt = 90, diag = FALSE,col = rev(COL2('RdBu')),is.corr = FALSE)  
dev.off();  
eoh=paste("Correlations with Full Plot of Female\_voek",n\_level,order,range,corre,method,".jpg")  
daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
corrplot(resultaf, type = "lower", order = order,method=method,tl.col = "black", tl.srt = 90, diag = FALSE,col = rev(COL2('RdBu')),is.corr = FALSE)  
  
jpeg(paste("Correlations with Full Plot of Male\_voeka",n\_level,order,range,corre,method,".jpg"), width = 8000, height = 8000, quality = 100,pointsize = 23, res=300);  
corrplot(resultam, type = "lower", order = order, method=method,tl.col = "black", tl.srt = 90, diag = FALSE,col = rev(COL2('RdBu')),is.corr = FALSE) #order = "alphabet", order = "hclust",  
dev.off();  
eoh=paste("Correlations with Full Plot of Male\_voeka",n\_level,order,range,corre,method,".jpg")  
daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
corrplot(resultam, type = "lower", order = order, method=method,tl.col = "black", tl.srt = 90, diag = FALSE,col = rev(COL2('RdBu')),is.corr = FALSE) #order = "alphabet", order = "hclust",  
  
#The ok ones:  
x1=colnames(resulta)[c(1:6)]  
x1=c(x1[3:6],x1[1],x1[2])#x1[2],  
x5=x5[!x5=='Perfluorodecyl.ethanoic.acid']  
colnames(resulta)[colnames(resulta)=="17aOH-P4"]="17a-OHP4"  
colnames(p.mat.BMI\_ordered\_MASLD)[colnames(p.mat.BMI\_ordered\_MASLD)=="17aOH-P4"]="17a-OHP4"  
x2[x2=="17aOH-P4"]="17a-OHP4"  
x2=x2[order(match(x2,steroidGroups[,2]))] #https://stackoverflow.com/questions/1568511/how-do-i-sort-one-vector-based-on-values-of-another  
x5=x5[!x5=='Perfluorodecyl.ethanoic.acid']  
#  
# resulta1=resulta[c(x1,x2),x5];p.mat.a1=p.mat.aa[c(x1,x2),x5]  
# resulta2=resultaf[c(x1,x2),x5];p.mat.f1=p.mat.ff[c(x1,x2),x5]  
# resulta3=resultam[c(x1,x2),x5];p.mat.m1=p.mat.mm[c(x1,x2),x5]  
# # tv\_ah=rango(resulta3,(min(resulta2)),max(resulta2)); resulta3=tv\_ah;#  
# hip1='transposesa\_kaikki scale';width = 2400;height=6000;pch.cex=1.2;  
# ho='PFAS vs. clinical factors and steroids'  
# resulta1=t(resulta1);resulta2=t(resulta2);resulta3=t(resulta3)  
# p.mat.a1=t(p.mat.a1);p.mat.f1=t(p.mat.f1);p.mat.m1=t(p.mat.m1)  
# width = 6000;height=2800;  
#  
resulta1=resulta[c(x3,x6),x5];p.mat.a1=p.mat.aa[c(x3,x6),x5]  
resulta2=resultaf[c(x3,x6),x5];p.mat.f1=p.mat.ff[c(x3,x6),x5]  
resulta3=resultam[c(x3,x6),x5];p.mat.m1=p.mat.mm[c(x3,x6),x5]  
# tv\_ah=rango(resulta3,(min(resulta2)),max(resulta2)); resulta3=tv\_ah;#  
hip1='transpose';width = 2400;height=6000;pch.cex=1.2;ho='PFAS vs. BAs and lipids'  
resulta1=t(resulta1);resulta2=t(resulta2);resulta3=t(resulta3)  
p.mat.a1=t(p.mat.a1);p.mat.f1=t(p.mat.f1);p.mat.m1=t(p.mat.m1)  
width = 9000;height=2800;  
  
resulta1[resulta1 >0.4] = 0.4  
resulta1[resulta1 < -0.4] = -0.4  
  
resulta2[resulta2 >0.4] = 0.4  
resulta2[resulta2 < -0.4] = -0.4  
  
resulta3[resulta3 >0.4] = 0.4  
resulta3[resulta3 < -0.4] = -0.4  
  
# hist(as.numeric(unlist(resulta1)),breaks=30,ylim=c(0.0,40)) #xlim=c(0.04,0.4),  
# resulta1[resulta1 > 1] = 1  
# resulta1[resulta1 < -1] = -1 #col.lim=c(-0.4,0.4))  
#  
# #https://www.rdocumentation.org/packages/corrplot/versions/0.92/topics/corrplot  
# #https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html  
# #https://statisticsglobe.com/change-font-size-corrplot-r  
# #order can be: alphabet, hclust, original #https://stackoverflow.com/questions/51115495/how-patientNumbers-keep-order-of-the-correlation-plot-labels-as-same-in-the-datafile  
#  
order="original"; range='orig';corre='no\_rendorm'; type='full'; method='color';ga='All';gf='Female';gm='Male' #color square  
col = colorRampPalette(c('blue', 'white','orange'), alpha = TRUE)(100)  
cl.offset=1.0;cl.length=5;cl.cex = 1.3;pch.cex=1.3;pch=20;cl.pos = 'r';#cl.pos = 'BMI\_ordered\_NAFLD' ;#pch.cex=0.95,1.3; height=6300; pos 'BMI\_ordered\_NAFLD' cl.pos = 'BMI\_ordered\_NAFLD'  
jpeg(paste("Square Correlation Plot ofdd",ho,ga,hip1,"3.jpg"), width = width, height = height, quality = 100,pointsize = 30, res=300);# par( ps=ps)# par(cex.lab=90)  
corrplot(resulta1, type = type, order = order,method=method, p.mat=p.mat.a1, tl.col = "black", #sum(COL2('RdBu')=="#FF7417")  
 cl.cex = cl.cex, pch.cex=pch.cex, pch.col='black',pch=pch,#pitikÃ¶ vain pch lisÃ¤tÃ¤ pch vÃ¤riin vÃ¤riin... mystistÃ¤...'#FEE12B'  
sig.level = c(.001,.05, .2),cl.pos = cl.pos, insig = "label\_sig", cl.offset=cl.offset,cl.length=cl.length,  
tl.srt = 90, diag = TRUE,col = col,is.corr = FALSE, col.lim=c(-0.4,0.4) ) #only in age...0.001,  
dev.off();eoh=paste("Square Correlation Plot ofdd",ho,ga,hip1,"3.jpg")  
daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
# pch.cex=1.3;  
jpeg(paste("Square Correlation Plot ofd",ho,gf,hip1,"3.jpg"), width = width, height = height, quality = 100,pointsize = 30, res=300);  
corrplot(resulta2, type = type, order = order,method=method, p.mat=p.mat.f1,tl.col = "black",  
 cl.cex = cl.cex, pch.cex=pch.cex,pch.col='black',pch=pch,  
sig.level = c(.001, .05, .2), cl.pos = cl.pos, insig = "label\_sig",cl.offset=cl.offset,cl.length=cl.length,  
tl.srt = 90, diag = TRUE,col = col,is.corr = FALSE,col.lim=c(-0.4,0.4)) #  
dev.off();eoh=paste("Square Correlation Plot ofd",ho,gf,hip1,"3.jpg")  
daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
# pch.cex=2.9;  
jpeg(paste("Square Correlation Plot ofd",ho,gm,hip1,"3.jpg"), width = width, height = height, quality = 100,pointsize = 30, res=300);  
corrplot(resulta3, type = type, order = order,method=method, p.mat=p.mat.m1, tl.col = "black", cl.cex = cl.cex,pch.cex=pch.cex,  
 pch.col='black',pch=pch,  
sig.level = c(.001, .05, .2),cl.pos = cl.pos, insig = "label\_sig",cl.offset=cl.offset,cl.length=cl.length,  
tl.srt = 90, diag = TRUE,col = col,is.corr = FALSE,col.lim=c(-0.4,0.4)) #,is.corr = FALSE  
dev.off();eoh=paste("Square Correlation Plot ofd",ho,gm,hip1,"3.jpg")  
daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
#  
#  
resulta1=resulta[c(x1,x3,x6),x2]; p.mat.a1=p.mat.aa[c(x1,x3,x6),x2]  
resulta2=resultaf[c(x1,x3,x6),x2]; p.mat.f1=p.mat.ff[c(x1,x3,x6),x2]  
resulta3=resultam[c(x1,x3,x6),x2]; p.mat.m1=p.mat.mm[c(x1,x3,x6),x2]  
# tv\_ah=rango(resulta3,(min(resulta2)),max(resulta2)); resulta3=tv\_ah;#  
hip1='transpose'; width = 3700;height=6300;ho='steroids vs. all others except PFAS';ps=28 #pch=10;  
# min(c(resulta2)); max(c(resulta2)) #These are around -0.4 and 0.4  
col = colorRampPalette(c('blue', 'white','orange'), alpha = TRUE)(100)  
  
# path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; setwd(path)  
resulta1[resulta1 >0.5] = 0.5  
resulta1[resulta1 < -0.5] = -0.5  
  
resulta2[resulta2 >0.5] = 0.5  
resulta2[resulta2 < -0.5] = -0.5  
  
resulta3[resulta3 >0.5] = 0.5  
resulta3[resulta3 < -0.5] = -0.5  
#  
#  
# resulta1=resulta[c(x3,x6),x2]; p.mat.a1=p.mat.aa[c(x3,x6),x2]  
# resulta2=resultaf[c(x3,x6),x2]; p.mat.f1=p.mat.ff[c(x3,x6),x2]  
# resulta3=resultam[c(x3,x6),x2]; p.mat.m1=p.mat.mm[c(x3,x6),x2]  
# # tv\_ah=rango(resulta3,(min(resulta2)),max(resulta2)); resulta3=tv\_ah;#  
# resulta1=t(resulta1);resulta2=t(resulta2);resulta3=t(resulta3);p.mat.a1=t(p.mat.a1);p.mat.f1=t(p.mat.f1);p.mat.m1=t(p.mat.m1);  
#  
# # resulta1[resulta1 > 0.25] = 0.4  
# # resulta1[resulta1 < -0.25] = -0.4  
# hist(as.numeric(unlist(resulta1)),breaks=30,ylim=c(0.0,40)) #xlim=c(0.04,0.4),  
#  
hip1='transpose'; width = 3200;height=2000;ho='steroids vs. all others except PFAS';ps=12 #pch=10;  
min(c(resulta2)); max(c(resulta2)) #These are around -0.4 and 0.4  
col = colorRampPalette(c('blue', 'white','orange'), alpha = TRUE)(150)  
  
# resulta1=resulta1[,steroidGroups[,'Abbreviation']];resulta2=resulta2[,steroidGroups[,'Abbreviation']];resulta3=resulta3[,steroidGroups[,'Abbreviation']]  
  
order="original"; range='orig';corre='no\_renorm'; type='full'; method='color';ga='All';gf='Female';gm='Male' #color square  
cl.offset=1.0;cl.length=5;cl.cex = 1.0;pch.cex=1.0;pch=20;cl.pos = 'r';#cl.pos = 'BMI\_ordered\_NAFLD' ;#pch.cex=0.95,1.3; height=6300; pos 'BMI\_ordered\_NAFLD' cl.pos = 'BMI\_ordered\_NAFLD'  
jpeg(paste("Square Correlation Plot ofrra",ho,ga,hip1,"3.jpg"), width = width, height = height, quality = 100,pointsize = 14, res=300);# par( ps=ps)# par(cex.lab=90)  
corrplot(resulta1, type = type, order = order,method=method, p.mat=p.mat.a1, tl.col = "black", #sum(COL2('RdBu')=="#FF7417")  
 cl.cex = cl.cex, pch.cex=pch.cex, pch.col='black',pch=pch,#pitikÃ¶ vain pch lisÃ¤tÃ¤ pch vÃ¤riin vÃ¤riin... mystistÃ¤...'#FEE12B'  
sig.level = c(.001,.05, .2),cl.pos = cl.pos, insig = "label\_sig", cl.offset=cl.offset,cl.length=cl.length,  
tl.srt = 90, diag = TRUE,col = col,is.corr = FALSE,col.lim=c(-0.5,0.5)) #only in age...0.001, is.corr = TRUE/FALSE rev(COL2('RdBu')[1:(length(COL2('RdBu'))-0)])  
dev.off();eoh=paste("Square Correlation Plot ofrr",ho,ga,hip1,"3.jpg")  
daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
  
pch.cex=1.3;  
jpeg(paste("Square Correlation Plot ofa",ho,gf,hip1,"3.jpg"), width = width, height = height, quality = 100,pointsize = 16, res=300);  
corrplot(resulta2, type = type, order = order,method=method, p.mat=p.mat.f1,tl.col = "black",  
 cl.cex = cl.cex, pch.cex=pch.cex,pch.col='black',pch=pch,  
sig.level = c(.001, .05, .2), cl.pos = cl.pos, insig = "label\_sig",cl.offset=cl.offset,cl.length=cl.length,  
tl.srt = 90, diag = TRUE,col = col,,is.corr = FALSE,col.lim=c(-0.5,0.5)) #  
dev.off();eoh=paste("Square Correlation Plot of",ho,gf,hip1,"3.jpg")  
daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
  
# pch.cex=2.9;  
jpeg(paste("Square Correlation Plot ofa",ho,gm,hip1,"3.jpg"), width = width, height = height, quality = 100,pointsize = 16, res=300);  
corrplot(resulta3, type = type, order = order,method=method, p.mat=p.mat.m1, tl.col = "black", cl.cex = cl.cex,pch.cex=pch.cex,  
 pch.col='black',pch=pch,  
sig.level = c(.001, .05, .2),cl.pos = cl.pos, insig = "label\_sig",cl.offset=cl.offset,cl.length=cl.length,  
tl.srt = 90, diag = TRUE,col = col,,is.corr = FALSE,col.lim=c(-0.5,0.5)) #,is.corr = FALSE  
dev.off();eoh=paste("Square Correlation Plot of",ho,gm,hip1,"3.jpg")  
daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
  
#Tiedoks: Copilotilla ei jÃ¤rkee edelliseen, vaikka nÃ¤yttÃ¤Ã¤ hieman siistimmÃ¤ltÃ¤.  
  
```  
  
<!-- {r, echo=FALSE, out.width="50%", fig.cap="Square Correlation Plot of steroids vs. all others except PFAS",fig.align="left"} -->  
<!-- knitr::include\_graphics("Square Correlation Plot ofrr steroids vs. all others except PFAS All transpose 3.jpg .svg") -->  
  
  
# Making Heatmaps with Linear Model Estimates  
```{r fig.align="left", fig.width=7.0, message=FALSE, warning=FALSE}  
  
# You may need BMI\_ordered\_MASLD rather big function patientNumbers calculate the estimates and plot at the same time, since the spaces of exper. interest have been reduced from the max dataset size.  
  
# Eli maksimilla vedetÃ¤Ã¤n... eli pitÃ¤is olla ok, sillÃ¤ skaalattu vastaa korrelaatiota tss. skaalaus vielÃ¤ miehiin...  
CalculateEstimates=function(CombinedData,Group,ok,fn,adj,sig.level,sick,sick\_group,joo) { # ok,aa,bb  
 CombinedData <- CovariatesScaledData  
 if (Group == 'male') {  
 NAFLDo <- CombinedData[CombinedData[, 'Gender'] == max(CombinedData[, 'Gender']), ]  
 } else if (Group == 'female') {  
 NAFLDo <- CombinedData[CombinedData[, 'Gender'] == min(CombinedData[, 'Gender']), ]  
 } else if (Group == 'All') {  
 NAFLDo <- CombinedData  
 }  
   
 SG0 <- NAFLDo[, c(2:dim(CombinedData)[2])]  
 columnNames <- colnames(SG0)  
 SG0 <- data.frame(SG0)  
 colnames(SG0[, 8:27]) <- gsub("-", ".", colnames(SG0[, 8:27]))  
 colnames(SG0[, 8:27]) <- gsub("/", ".", colnames(SG0[, 8:27]))  
   
 hesh <- data.frame()  
   
 TreatmentVariables <- colnames(AllData)[52:58]  
 MediatorVariables <- colnames(AllData)[9:28]  
 OutcomeVariables <- colnames(AllData)[c(29:51, 59:71)]  
 Treatment2 <- TreatmentVariables # Define Treatment2  
   
 xnam\_list <- list(colnames(SG0)[c(4:7)], colnames(SG0)[c(2)], colnames(SG0)[c(3)], Treatment2, c(x3, x6), c('AGE', 'BMI', colnames(SG0)[c(4:7)]), c(x3, x6))  
 y\_list <- list(Treatment2, Treatment2, Treatment2, colnames(SG0[, 8:27]), colnames(SG0[, 8:27]), colnames(SG0[, 8:27]), colnames(AllData)[52:58])  
   
 for (k in 1:length(xnam\_list)) {  
 xnam <- xnam\_list[[k]]  
 y <- y\_list[[k]]  
 for (i in 1:length(xnam)) {  
 for (j in 1:length(y)) {  
 if (Group != 'All') {  
 fmla <- as.formula(paste(paste(c(y[j], " ~ "), collapse = ""), paste(c(xnam[i], 'BMI', 'AGE'), collapse = "+")))  
 } else if (Group == 'All') {  
 fmla <- as.formula(paste(paste(c(y[j], " ~ "), collapse = ""), paste(c(xnam[i], 'BMI', 'AGE', 'Gender'), collapse = "+")))  
 }  
 poissone <- lm(fmla, data = SG0)  
 pss <- summary(poissone)[[4]]  
 hoesh <- c(y[j], xnam[i], Group, pss[2, 1], pss[2, 4], pss[2, 2])  
 hesh <- rbind(hesh, hoesh)  
 }  
 }  
 }  
   
 hesh <- as.data.frame(hesh)  
 colnames(hesh) <- c('y', 'x', 'Gender', 'r', 'p', 'var\_x')  
   
 hesa=hesh  
 hoi=as.data.frame(hesh)  
   
   
 hopiu=hoi  
 colnames(hopiu)=c('y','x','Gender','r','p','var\_x')  
 colnames(hoi)=c('y','x','Gender','r','p','var\_x')  
   
 #This in case you want patientNumbers print patientNumbers your local computer: ... :)  
 # main\_dir <- paste0(c("C://Users//patati//Desktop//Turku//R//",fn),collapse="")  
 main\_dir <- paste0(c("C://Users//patati//Documents//GitHub//Steroid\_Data\_Analysis//"),collapse="")  
 setwd(main\_dir)  
   
 meds=names(table(hoi[,1]))[!names(table(hoi[,1])) %in% c(x3,x5,x6)]  
 covas=c('Steatosis.Grade','Fibrosis.Stage','Necroinflammation','HOMA.IR','AGE','BMI')  
   
 if (adj=='ok') {  
 # p.adjust(p=hopiu[,5], method = 'BH', n = length(hopiu[,5]))  
 hoi[,5]=p.adjust(p=hopiu[,5], method = 'BH', n = length(hopiu[,5]))  
 hopiu[,5]=p.adjust(p=hopiu[,5], method = 'BH', n = length(hopiu[,5]))  
 }  
   
 if (ok=='big') {  
   
 rsa=c();joi=c()  
   
 # Eli 'kaksi' vielÃ¤ tarvitaan...  
 # 1) BA/lipid=covar ja steroidit, ja 2) steroid=covar  
  
 meds=names(table(hoi[,1]))[!names(table(hoi[,1])) %in% c(x3,x5,x6)]  
 covas=c('Steatosis.Grade','Fibrosis.Stage','Necroinflammation','HOMA.IR','AGE','BMI')  
   
 c1=hoi[,2] %in% covas  
 c2=hoi[,1] %in% meds  
 hyy=c1 & c2  
 m1=hoi[hyy,]  
 colnames(m1)=c('y','x','Gender','r','p','var\_x') #c('y','x','Gender','r','p','radj')  
 c1=hoi[,2] %in% c(x3,x6); c2=hoi[,1] %in% meds  
 hyy=c1 & c2; m2=hoi[hyy,]  
 colnames(m2)=c('y','x','Gender','r','p','var\_x') # hist(as.numeric(m2[,6]),breaks=50)  
 joi=rbind(m1,m2)  
 i=4;rs=c()  
   
 for (i in 4:5) {  
 rs=joi[,c(1,2,i)] # rs=data.frame(rs)  
 rs=reshape(rs,idvar="x",timevar="y",direction="wide")  
 rownames(rs)=rs[,1]  
 rs=rs[,-1]  
  
 library(stringr)  
 colnames(rs)=str\_sub(colnames(rs),3,-1)  
   
 colnames(rs) <- gsub("\\.", "-", colnames(rs))  
 colnames(rs) <- gsub("X11", "11", colnames(rs))  
 colnames(rs) <- gsub("X17", "17", colnames(rs))  
 colnames(rs)[colnames(rs)=="T-Epi-T"]="T/Epi-T"  
   
 rownames(rs)[rownames(rs)=="Steatosis.Grade"]="Steatosis Grade"  
 rownames(rs)[rownames(rs)=="Fibrosis.Stage"]="Fibrosis Stage"  
 rownames(rs)[rownames(rs)=="HOMA.IR"]="HOMA-IR"  
 covas[covas=="Steatosis.Grade"]="Steatosis Grade"  
 covas[covas=="Fibrosis.Stage"]="Fibrosis Stage"  
 covas[covas=="HOMA.IR"]="HOMA-IR"  
   
 heps=c(steroidGroups[,2]) #check that you have driven the steroid data vis file...  
 heps[heps=="17aOH-P4"]="17a-OHP4"  
 cme1=match(heps,colnames(rs))  
 cme2=match(c(covas,x3,x6),rownames(rs))  
 rs=rs[cme2,cme1]  
 rsa=rbind(rsa,rs) }  
   
   
 rs1a=rsa[1:dim(rs)[1],];  
 rs2a=rsa[(dim(rs1a)[1]+1):(dim(rs1a)[1]+dim(rs1a)[1]),]  
   
 rs1=rs1a;rs2=rs2a  
 rownames(rs2)=str\_sub(rownames(rs2), end = -2)  
 rownames(rs1) <- gsub("\\.", " ", rownames(rs1))  
 rownames(rs2) <- gsub("\\.", " ", rownames(rs2))  
 rownames(rs1)[rownames(rs1)=="HOMA IR"]="HOMA-IR";rownames(rs2)[rownames(rs2)=="HOMA IR"]="HOMA-IR"  
   
 rownames(rs1)[rownames(rs1)=="Gender"]="HOMA-IR";rownames(rs2)[rownames(rs2)=="HOMA IR"]="HOMA-IR"  
   
 rango = function(x,mi,ma) {(ma-mi)/(max(x)-min(x))\*(x-min(x))+mi}  
 rs1 <- mutate\_all(rs1, function(x) as.numeric(as.character(x)))  
 rs2 <- mutate\_all(rs2, function(x) as.numeric(as.character(x)))  
 # rs1=rango(rs1,-0.5,0.5) #check this if needed  
   
 rs1=as.matrix(rs1)  
 rs2=as.matrix(rs2)  
   
 rs1[rs1>0.5]=0.5;rs1[rs1 < -0.5]=-0.5;  
  
   
 width=2500; height=4400  
 order="original"; range='orig';corre='no\_renorm'; type='full'; method='color';#ga='All';gf='Female';gm='Male' #color square  
 cl.offset=1.0;cl.length=15;cl.cex = 1.09;pch.cex=1.09;pch=11;cl.pos = 'n';#cl.pos = 'BMI\_ordered\_NAFLD' ;#pch.cex=0.95,1.3; height=6300; pos 'BMI\_ordered\_NAFLD' cl.pos = 'BMI\_ordered\_NAFLD'  
 ho=Group;hip1='BAs\_lipids\_as\_y vs. steroids\_as\_x'  
  
 # https://www.rdocumentation.org/packages/corrplot/versions/0.94/topics/corrplot  
 # Oh! https://www.tidyverse.org/blog/2020/08/taking-control-of-plot-scaling/  
 # https://r4ds.had.co.nz/graphics-for-communication.html#figure-sizing  
  
#I have driven these separately for html:   
 # for that you need:   
 jpeg(paste("Linear Model Estimate Plot ofees5",hip1,Group,".jpg"), width = width, height = height, quality = 100,pointsize = 16, res=300);  
  
 hepio=colorRampPalette(c('blue', 'white','orange'), alpha = TRUE)(150) #rev(COL2('RdBu')[25:(length(COL2('RdBu'))-25)])  
   
   
   
 corrplot(rs1, type = type, order = order,method=method, p.mat=rs2, tl.col = "black", cl.cex = cl.cex,pch.cex=pch.cex,pch.col='black',pch=pch, ,sig.level = c(.001, .01, .05),cl.pos = cl.pos,   
 insig = "label\_sig",cl.offset=cl.offset,cl.length=cl.length,tl.cex=0.5, tl.srt = 90, diag = TRUE, #tl.pos='n'  
 col = colorRampPalette(c('blue', 'white','orange'), alpha = TRUE)(100) ,is.corr = FALSE,col.lim=c(-0.5,0.5)); #https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html#change-color-spectra-color-legend-and-text-legend  
   
 dev.off() #,is.corr = FALSE  
 eoh=paste("Linear Model Estimate Plot ofees5",hip1,Group,".jpg")  
 daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
 my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
  
 #  
  
 #https://stackoverflow.com/questions/26574054/how-patientNumbers-change-font-size-of-the-correlation-coefficient-in-corrplot  
 #https://stackoverflow.com/questions/9543343/plot-BMI\_ordered\_MASLD-jpg-image-using-base-graphics-in-r  
 #oh, classical: https://forum.posit.co/t/r-markdown-html-document-doesnt-show-image/41629/2  
  
   
 } else {  
   
 rsa=c();rs1=c();rs2=c()  
   
 c1=hoi[,1] %in% x5  
 hoi[c1,] # This gives you the PFAS (x5) ok. TÃ¤Ã¤ tulee suoraan tÃ¤llÃ¤ PFAS 7:lle ekalle  
 c2=hoi[,1] %in% names(table(hoi[,1]))[!names(table(hoi[,1])) %in% c(x3,x5,x6)] #vaikeemman kautta  
 hyy=c1 & c2  
 hoi2=hoi[c2 | c1 ,] #Likewise...  
 rownames(hoi2)=1:dim(hoi2)[1]  
 hoi2=hoi2[1:182,]  
 BMI\_ordered\_MASLD=hoi2[1:42,1];BMI\_ordered\_NAFLD=hoi2[1:42,2]  
 hoi2[1:42,]=cbind(BMI\_ordered\_NAFLD,BMI\_ordered\_MASLD,hoi2[1:42,3:5])   
 hoi2=hoi2[,c(2,1,3:5)]   
   
 i=4;  
 rse=c()  
 for (i in 4:5) {  
   
 rse=hoi2[,c(1,2,i)]   
 rse=rse[order(rse[,1]),]  
 rs=reshape(rse,idvar="x",timevar="y",direction="wide")  
 rownames(rs)=rs[,1]  
 rs=rs[,-1]  
   
 library(stringr)   
 colnames(rs)=str\_sub(colnames(rs),3,-1)  
  
 colnames(rs) <- gsub("\\.", "-", colnames(rs))  
 colnames(rs) <- gsub("X11", "11", colnames(rs))  
 colnames(rs) <- gsub("X17", "17", colnames(rs))  
 colnames(rs)[colnames(rs)=="T-Epi-T"]="T/Epi-T"  
 colnames(rs)[colnames(rs)=="T-E-T"]="T/Epi-T"  
 colnames(rs)[colnames(rs)=="Steatosis-Grade"]="Steatosis Grade"  
 colnames(rs)[colnames(rs)=="Fibrosis-Stage"]="Fibrosis Stage"  
 colnames(rs)[colnames(rs)=="17aOH-P4"]="17a-OHP4"  
 heps=c(covas,steroidGroups[,2])  
 heps <- gsub("\\.", " ", heps)  
 heps[heps=="HOMA IR"]="HOMA-IR"  
 heps[heps=="17aOH-P4"]="17a-OHP4"  
 ccc=match(heps,colnames(rs))  
  
 rs=rs[,ccc]  
 rsa=rbind(rsa,rs)   
 }  
   
 rs1a=rsa[1:7,];  
 rs2a=rsa[8:14,]  
 rs1=rs1a;rs2=rs2a  
   
   
  
 rs1 <- mutate\_all(rs1, function(x) as.numeric(as.character(x)))  
 rs2 <- mutate\_all(rs2, function(x) as.numeric(as.character(x)))  
 # rs1=rango(rs1,-0.5,0.5) #check if needed  
  
 rs1=as.matrix(rs1)  
 rs2=as.matrix(rs2)  
   
 rs1[rs1>0.4]=0.4;rs1[rs1 < -0.4]=-0.4;  
  
 order="original"; range='orig';corre='no\_renorm'; type='full'; method='color'; #ga='All';gf='Female';gm='Male' #color square  
 cl.offset=1.0;cl.length=11;cl.cex = 1.4;pch.cex=1.5;pch=20;cl.pos = 'n';#cl.pos = 'BMI\_ordered\_NAFLD' ;#pch.cex=0.95,1.3; height=6300; pos 'BMI\_ordered\_NAFLD' cl.pos = 'BMI\_ordered\_NAFLD'  
 ho=Group;hip1='Steroids\_y vs. PFAS\_as\_x'  
 width=5500; height=1800  
#I have driven these separately for html:   
  
   
   
   
jpeg(paste("Linear Model Estimate Plot of\_sa5",hip1,Group,".jpg"), width = width, height = height, quality = 100,pointsize = 16, res=300);  
 corrplot(rs1, type = type, order = order,method=method, p.mat=rs2, tl.col = "black", cl.cex = cl.cex,pch.cex=pch.cex,pch.col='black',pch=pch,  
sig.level = c(.001, .01, .05),cl.pos = cl.pos, insig = "label\_sig",cl.offset=cl.offset,cl.length=cl.length, tl.cex=0.8, #tl.pos='n',  
tl.srt = 90, diag = TRUE,col = colorRampPalette(c('blue','white', 'orange'), alpha = TRUE)(100),is.corr = FALSE,col.lim=c(-0.4,0.4));   
  
# https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html#change-color-spectra-color-legend-and-text-legend  
dev.off();  
 eoh=paste("Linear Model Estimate Plot of\_sa5",hip1,Group,".jpg")  
 daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
 my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
  
  
 }   
   
  
 return(list(hopiu))  
}  
  
# The scaling here just in case:  
rango = function(x,mi,ma) {(ma-mi)/(max(x)-min(x))\*(x-min(x))+mi}  
  
#To apply patientNumbers all steroidGroups at one go:  
RunAnalysis=function(CombinedData,adj,sig.level,sick,sick\_group,joo) {  
 RunAnalysis=c();huusa=c();heijaa=c('All','female','male'); ok=c('big','small') ; jj=c()  
 hyp=1;hrt=1;#oo="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/lme/"  
 for (hyp in 1:2) {  
 for (hrt in 1:3) {  
 # CalculateEstimates=function(CombinedData,Group,ok,aa,bb,fn,adj)  
 RunAnalysis=append(RunAnalysis,CalculateEstimates(CombinedData,heijaa[hrt],ok[hyp],fn,adj,sig.level,sick,sick\_group,joo))}}  
  
 return(RunAnalysis)}  
  
# Driving the function with the parameters as follows:  
adj='nook'; sig.level=c(.001,0.01, 0.05); sick='no'; joo='joo' #sickGroup..  
metanorm\_S\_non\_fdr=RunAnalysis(AllData,adj,sig.level,sick,sick\_group,joo) #This prints patientNumbers your current folder 'Steroid\_Data\_Analysis  
  
# Ei mee tÃ¤Ã¤kÃ¤n ihan nopeesti lÃ¤pi ChatGPT:llÃ¤ saati Copilotkaan.  
# TÃ¤Ã¤ olis kiva saada paremmaks, joten jÃ¤tÃ¤n tÃ¤mÃ¤n OS. (Funktiota.R lÃ¶ytyy kansiosta, siellÃ¤ lisÃ¤Ã¤, yks minitoimiva on.)  
  
  
```  
  
```{r, echo=FALSE, out.width="50%", fig.cap="Heatmap of LMEs (linear model estimates) from Steroids vs. BAs & Lipids & Covariates with All Subjects",fig.align="left"}  
knitr::include\_graphics("Linear Model Estimate Plot ofees5 BAs\_lipids\_as\_y vs. steroids\_as\_x All .jpg")  
```  
```{r, echo=FALSE, out.width="50%", fig.cap="Heatmap of LMEs from Steroids vs. BAs & Lipids & Covariates with Female Subjects",fig.align="left"}  
knitr::include\_graphics("Linear Model Estimate Plot ofees5 BAs\_lipids\_as\_y vs. steroids\_as\_x female .jpg")  
```  
```{r, echo=FALSE, out.width="50%", fig.cap="Heatmap of LMEs from Steroids vs. BAs & Lipids & Covariates with Male Subjects",fig.align="left"}  
knitr::include\_graphics("Linear Model Estimate Plot ofees5 BAs\_lipids\_as\_y vs. steroids\_as\_x male .jpg")  
```  
```{r, echo=FALSE, out.width="50%", fig.cap="Heatmap of LMEs from Steroids vs. PFAS & Covariates with All Subjects",fig.align="left"}  
knitr::include\_graphics("Linear Model Estimate Plot of\_sa5 Steroids\_y vs. PFAS\_as\_x All .jpg")  
```  
```{r, echo=FALSE, out.width="50%", fig.cap="Heatmap of LMEs from Steroids vs. PFAS & Covariates with Female Subjects",fig.align="left"}  
knitr::include\_graphics("Linear Model Estimate Plot of\_sa5 Steroids\_y vs. PFAS\_as\_x female .jpg")  
```  
```{r, echo=FALSE, out.width="50%", fig.cap="Heatmap of LMEs from Steroids vs. PFAS & Covariates with Male Subjects",fig.align="left"}  
#fig.cap="Linear Model Estimates between Steroids and Variables"  
knitr::include\_graphics("Linear Model Estimate Plot of\_sa5 Steroids\_y vs. PFAS\_as\_x male .jpg")  
# [1] "Linear Model Estimate Plot of BAs\_lipids\_as\_y vs. steroids\_as\_x All .jpg"   
# [2] "Linear Model Estimate Plot of BAs\_lipids\_as\_y vs. steroids\_as\_x female .jpg"  
# [3] "Linear Model Estimate Plot of BAs\_lipids\_as\_y vs. steroids\_as\_x male .jpg"   
# [4] "Linear Model Estimate Plot of Steroids\_y vs. PFAS\_as\_x All .jpg"   
# [5] "Linear Model Estimate Plot of Steroids\_y vs. PFAS\_as\_x female .jpg"   
# [6] "Linear Model Estimate Plot of Steroids\_y vs. PFAS\_as\_x male .jpg"   
```  
  
  
  
  
# Making Scatter Plots with Models   
```{r, warning=FALSE,message=FALSE,fig.width=10.0,fig.align="left"}  
# This should be as good as it gets... eli maksimilla vedetÃ¤Ã¤n... eli pitÃ¤is olla ok, sillÃ¤ skaalattu vastaa korrelaatiota tss. skaalaus vielÃ¤ miehiin...  
# This looks similar as the linear model code, but is not exactly so. With this you can draw BMI\_ordered\_MASLD scatter plot for each individual combination of two variables within dataset  
CreateScatterPlots=function(AllData, Group, hopiu, aa, bb, fn) {  
 # Filter data based on the specified group  
 if (Group == 'male') {  
 NAFLDo = AllData[AllData[,'Gender'] == max(AllData[,'Gender']),]  
 } else if (Group == 'female') {  
 NAFLDo = AllData[AllData[,'Gender'] == min(AllData[,'Gender']),]  
 } else if (Group == 'All') {  
 NAFLDo = AllData  
 }  
   
 # Select relevant columns from the filtered data  
 SG0 = NAFLDo[, c(2:dim(AllData)[2])]  
   
 # Fix column names by replacing spaces and special characters  
 columnNames = colnames(SG0)  
 SG0 = data.frame(SG0)  
 colnames(SG0[, 8:27]) <- gsub("-", ".", colnames(SG0[, 8:27]))  
 colnames(SG0[, 8:27]) <- gsub("/", ".", colnames(SG0[, 8:27]))  
   
 hesh = c()  
 xnam = colnames(SG0)[c(4:7)]  
 TreatmentVariables = colnames(AllData)[52:58]  
 y = TreatmentVariables  
 TreatmentN = TreatmentVariables  
   
 # Loop through each combination of xnam and y patientNumbers calculate correlations  
 for (i in 1:length(xnam)) {  
 for (j in 1:length(y)) {  
 # Extract relevant rows from hopiu based on conditions  
 hÃ¶sh = hopiu[hopiu[, 1] == y[j] & hopiu[, 2] == xnam[i] & hopiu[, 3] == Group,]  
   
 # Prepare data for plotting  
 jeps = SG0  
 r = as.numeric(hÃ¶sh[4][1,])  
 p.valueList = as.numeric(hÃ¶sh[5][1,])  
 rsadj = as.numeric(hÃ¶sh[6][1,])  
 colnames(jeps) = colnames(AllData)[2:dim(AllData)[2]]  
 TreatmentVariables = as.character(hÃ¶sh[2][1,])  
 MediatorVariables = as.character(hÃ¶sh[1][1,])  
   
 # Clean up MediatorVariables names  
 MediatorVariables <- gsub("\\.", "-", MediatorVariables)  
 MediatorVariables <- gsub("X", "", MediatorVariables)  
 if (MediatorVariables == "T-Epi-T") {  
 MediatorVariables[MediatorVariables == "T-Epi-T"] = "T/Epi-T"  
 }  
 Treatment2 = TreatmentVariables  
 colnames(jeps)[colnames(jeps) == 'HOMA-IR'] = 'HOMA.IR'  
 colnames(jeps) <- gsub(" ", "\\.", colnames(jeps))  
   
 # Set up directory for saving plots  
 main\_dir <- paste0(c("C://Users//patati//Desktop//Turku//R//", fn, Group, "/"), collapse = "")  
 sub\_dir <- TreatmentVariables  
 if (file.exists(file.path(main\_dir, sub\_dir))) {  
 setwd(file.path(main\_dir, sub\_dir))  
 } else {  
 dir.create(file.path(main\_dir, sub\_dir))  
 setwd(file.path(main\_dir, sub\_dir))  
 }  
   
 # Save plots as JPEG files  
 if (MediatorVariables != "T/Epi-T") {  
 jpeg(paste("Correlations plotted", Group, Treatment2, MediatorVariables, ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 } else {  
 jpeg(paste("Correlations plotted\_alle", Treatment2, 'T.Epi.T', ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 }  
   
 # Plot correlations using ggplot2  
 xcx = jeps[, TreatmentVariables]  
 ycy = jeps[, MediatorVariables]  
 vÃ¤h = round(sd(ycy) / 2, 2)  
 BMI\_ordered\_MASLD = ggplot(jeps, aes(y = ycy, x = xcx)) +  
 geom\_point() +  
 xlab(TreatmentVariables) +  
 ylab(MediatorVariables) +  
 geom\_smooth(method = "lm", col = "black") +  
 annotate("text", x = min(xcx), y = max(ycy), hjust = 0, vjust = 0, label = paste0("r = ", round(r, 5))) +  
 annotate("text", x = min(xcx), y = max(ycy) - vÃ¤h, hjust = 0, vjust = 0, label = paste0("p = ", round(p.valueList, 5))) +  
 theme\_classic()  
   
 print(BMI\_ordered\_MASLD)  
 dev.off()  
 }  
 }  
   
 # Repeat the process for different combinations of xnam and y  
 xnam <- colnames(SG0)[c(2)]  
 y <- TreatmentN  
 for (i in 1:length(xnam)) {  
 for (j in 1:length(y)) {  
 hÃ¶sh = hopiu[hopiu[, 1] == y[j] & hopiu[, 2] == xnam[i] & hopiu[, 3] == Group,]  
 jeps = SG0  
 r = as.numeric(hÃ¶sh[4][1,])  
 p.valueList = as.numeric(hÃ¶sh[5][1,])  
 rsadj = as.numeric(hÃ¶sh[6][1,])  
 colnames(jeps) = colnames(AllData)[2:dim(AllData)[2]]  
 TreatmentVariables = as.character(hÃ¶sh[2][1,])  
 MediatorVariables = as.character(hÃ¶sh[1][1,])  
 MediatorVariables <- gsub("\\.", "-", MediatorVariables)  
 MediatorVariables <- gsub("X", "", MediatorVariables)  
 if (MediatorVariables == "T-Epi-T") {  
 MediatorVariables[MediatorVariables == "T-Epi-T"] = "T/Epi-T"  
 }  
 colnames(jeps)[colnames(jeps) == 'HOMA-IR'] = 'HOMA.IR'  
 colnames(jeps) <- gsub(" ", "\\.", colnames(jeps))  
 main\_dir <- paste0(c("C://Users//patati//Desktop//Turku//R//", fn, Group, "/"), collapse = "")  
 sub\_dir <- TreatmentVariables  
 if (file.exists(file.path(main\_dir, sub\_dir))) {  
 setwd(file.path(main\_dir, sub\_dir))  
 } else {  
 dir.create(file.path(main\_dir, sub\_dir))  
 setwd(file.path(main\_dir, sub\_dir))  
 }  
 if (MediatorVariables != "T/Epi-T") {  
 jpeg(paste("Correlations plotted", Group, Treatment2, MediatorVariables, ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 } else {  
 jpeg(paste("Correlations plotted", Treatment2, 'T.Epi.T', ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 }  
 xcx = jeps[, TreatmentVariables]  
 ycy = jeps[, MediatorVariables]  
 vÃ¤h = round(sd(ycy) / 2, 2)  
 BMI\_ordered\_MASLD = ggplot(jeps, aes(y = ycy, x = xcx)) +  
 geom\_point() +  
 xlab(TreatmentVariables) +  
 ylab(MediatorVariables) +  
 geom\_smooth(method = "lm", col = "black") +  
 annotate("text", x = min(xcx), y = max(ycy), hjust = 0, vjust = 0, label = paste0("r = ", round(r, 4))) +  
 annotate("text", x = min(xcx), y = max(ycy) - vÃ¤h, hjust = 0, vjust = 0, label = paste0("p = ", round(p.valueList, 5))) +  
 theme\_classic()  
 print(BMI\_ordered\_MASLD)  
 dev.off()  
 hesh = rbind(hesh, c(y[j], xnam[i], Group, r, p.valueList, rsadj))  
 }  
 }  
   
 # Repeat the process for different combinations of xnam and y  
 xnam <- colnames(SG0)[c(3)]  
 y <- TreatmentN  
 for (i in 1:length(xnam)) {  
 for (j in 1:length(y)) {  
 hÃ¶sh = hopiu[hopiu[, 1] == y[j] & hopiu[, 2] == xnam[i] & hopiu[, 3] == Group,]  
 jeps = SG0  
 r = as.numeric(hÃ¶sh[4][1,])  
 p.valueList = as.numeric(hÃ¶sh[5][1,])  
 rsadj = as.numeric(hÃ¶sh[6][1,])  
 colnames(jeps) = colnames(AllData)[2:dim(AllData)[2]]  
 TreatmentVariables = as.character(hÃ¶sh[2][1,])  
 MediatorVariables = as.character(hÃ¶sh[1][1,])  
 MediatorVariables <- gsub("\\.", "-", MediatorVariables)  
 MediatorVariables <- gsub("X", "", MediatorVariables)  
 if (MediatorVariables == "T-Epi-T") {  
 MediatorVariables[MediatorVariables == "T-Epi-T"] = "T/Epi-T"  
 }  
 Treatment2 = TreatmentVariables  
 colnames(jeps)[colnames(jeps) == 'HOMA-IR'] = 'HOMA.IR'  
 colnames(jeps) <- gsub(" ", "\\.", colnames(jeps))  
 main\_dir <- paste0(c("C://Users//patati//Desktop//Turku//R//", fn, Group, "/"), collapse = "")  
   
 sub\_dir <- TreatmentVariables  
 if (file.exists(file.path(main\_dir, sub\_dir))) {  
 setwd(file.path(main\_dir, sub\_dir))  
 } else {  
 dir.create(file.path(main\_dir, sub\_dir))  
 setwd(file.path(main\_dir, sub\_dir))  
 }  
 if (MediatorVariables != "T/Epi-T") {  
 jpeg(paste("Correlations plotted\_alle", Treatment2, MediatorVariables, ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 } else {  
 jpeg(paste("Correlations plotted", Treatment2, 'T.Epi.T', ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 }  
   
 xcx = jeps[, TreatmentVariables]  
 ycy = jeps[, MediatorVariables]  
 vÃ¤h = round(sd(ycy) / 2, 2)  
 BMI\_ordered\_MASLD = ggplot(jeps, aes(y = ycy, x = xcx)) +  
 geom\_point() +  
 xlab(TreatmentVariables) +  
 ylab(MediatorVariables) +  
 geom\_smooth(method = "lm", col = "black") +  
 annotate("text", x = min(xcx), y = max(ycy), hjust = 0, vjust = 0, label = paste0("r = ", round(r, 5))) +  
 annotate("text", x = min(xcx), y = max(ycy) - vÃ¤h, hjust = 0, vjust = 0, label = paste0("p = ", round(p.valueList, 5))) +  
 theme\_classic()  
 print(BMI\_ordered\_MASLD)  
 dev.off()  
 }  
 }  
   
 # Repeat the process for different combinations of xnam and y  
 xnam <- TreatmentN  
 y = colnames(SG0[, 8:27])  
 for (i in 1:length(xnam)) {  
 for (j in 1:length(y)) {  
 hÃ¶sh = hopiu[hopiu[, 1] == y[j] & hopiu[, 2] == xnam[i] & hopiu[, 3] == Group,]  
 jeps = SG0  
 r = as.numeric(hÃ¶sh[4][1,])  
 p.valueList = as.numeric(hÃ¶sh[5][1,])  
 rsadj = as.numeric(hÃ¶sh[6][1,])  
 colnames(jeps) = colnames(AllData)[2:dim(AllData)[2]]  
 TreatmentVariables = as.character(hÃ¶sh[2][1,])  
 MediatorVariables = as.character(hÃ¶sh[1][1,])  
 MediatorVariables <- gsub("\\.", "-", MediatorVariables)  
 MediatorVariables <- gsub("X", "", MediatorVariables)  
 if (MediatorVariables == "T-Epi-T") {  
 MediatorVariables[MediatorVariables == "T-Epi-T"] = "T/Epi-T"  
 }  
 Treatment2 = TreatmentVariables  
 colnames(jeps)[colnames(jeps) == 'HOMA-IR'] = 'HOMA.IR'  
 colnames(jeps) <- gsub(" ", "\\.", colnames(jeps))  
 main\_dir <- paste0(c("C://Users//patati//Desktop//Turku//R//", fn, Group, "/"), collapse = "")  
 sub\_dir <- TreatmentVariables  
 if (file.exists(file.path(main\_dir, sub\_dir))) {  
 setwd(file.path(main\_dir, sub\_dir))  
 } else {  
 dir.create(file.path(main\_dir, sub\_dir))  
 setwd(file.path(main\_dir, sub\_dir))  
 }  
 if (MediatorVariables != "T/Epi-T") {  
 jpeg(paste("Correlations plotted", Group, Treatment2, MediatorVariables, ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 } else {  
 jpeg(paste("Correlations plotted", Treatment2, 'T.Epi.T', ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 }  
 xcx = jeps[, TreatmentVariables]  
 ycy = jeps[, MediatorVariables]  
 vÃ¤h = round(sd(ycy) / 2, 2)  
 BMI\_ordered\_MASLD = ggplot(jeps, aes(y = ycy, x = xcx)) +  
 geom\_point() +  
 xlab(TreatmentVariables) +  
 ylab(MediatorVariables) +  
 geom\_smooth(method = "lm", col = "black") +  
 annotate("text", x = min(xcx), y = max(ycy), hjust = 0, vjust = 0, label = paste0("r = ", round(r, 4))) +  
 annotate("text", x = min(xcx), y = max(ycy) - vÃ¤h, hjust = 0, vjust = 0, label = paste0("p = ", round(p.valueList, 5))) +  
 theme\_classic()  
 print(BMI\_ordered\_MASLD)  
 dev.off()  
 }  
 }  
   
 # Additional analysis for specific combinations of xnam and y  
 xnam <- c(x3, x6)  
 y <- c(colnames(SG0[, 8:27]))  
 for (i in 1:length(xnam)) {  
 for (j in 1:length(y)) {  
 hÃ¶sh = hopiu[hopiu[, 1] == y[j] & hopiu[, 2] == xnam[i] & hopiu[, 3] == Group,]  
 jeps = SG0  
 r = as.numeric(hÃ¶sh[4][1,])  
 p.valueList = as.numeric(hÃ¶sh[5][1,])  
 rsadj = as.numeric(hÃ¶sh[6][1,])  
 colnames(jeps) = colnames(AllData)[2:dim(AllData)[2]]  
 TreatmentVariables = as.character(hÃ¶sh[2][1,])  
 MediatorVariables = as.character(hÃ¶sh[1][1,])  
 MediatorVariables <- gsub("\\.", "-", MediatorVariables)  
 MediatorVariables <- gsub("X", "", MediatorVariables)  
 if (MediatorVariables == "T-Epi-T") {  
 MediatorVariables[MediatorVariables == "T-Epi-T"] = "T/Epi-T"  
 }  
 Treatment2 = TreatmentVariables  
 colnames(jeps)[colnames(jeps) == 'HOMA-IR'] = 'HOMA.IR'  
 colnames(jeps) <- gsub(" ", "\\.", colnames(jeps))  
 main\_dir <- paste0(c("C://Users//patati//Desktop//Turku//R//", fn, Group, "/"), collapse = "")  
 sub\_dir <- TreatmentVariables  
 if (file.exists(file.path(main\_dir, sub\_dir))) {  
 setwd(file.path(main\_dir, sub\_dir))  
 } else {  
 dir.create(file.path(main\_dir, sub\_dir))  
 setwd(file.path(main\_dir, sub\_dir))  
 }  
 if (MediatorVariables != "T/Epi-T") {  
 jpeg(paste("Correlations plotted", Group, Treatment2, MediatorVariables, ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 } else {  
 jpeg(paste("Correlations plotted", Treatment2, 'T.Epi.T', ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 }  
 xcx = jeps[, TreatmentVariables]  
 ycy = jeps[, MediatorVariables]  
 vÃ¤h = round(sd(ycy) / 2, 2)  
 BMI\_ordered\_MASLD = ggplot(jeps, aes(y = ycy, x = xcx)) +  
 geom\_point() +  
 xlab(TreatmentVariables) +  
 ylab(MediatorVariables) +  
 geom\_smooth(method = "lm", col = "black") +  
 annotate("text", x = min(xcx), y = max(ycy), hjust = 0, vjust = 0, label = paste0("r = ", round(r, 5))) +  
 annotate("text", x = min(xcx), y = max(ycy) - vÃ¤h, hjust = 0, vjust = 0, label = paste0("p = ", round(p.valueList, 5))) +  
 theme\_classic()  
 print(BMI\_ordered\_MASLD)  
 dev.off()  
 hesh = rbind(hesh, c(y[j], xnam[i], Group, r, p.valueList, rsadj))  
 }  
 }  
   
 # Final analysis for specific combinations of xnam and y  
 xnam <- c('AGE', 'BMI', colnames(SG0)[c(4:7)])  
 y <- c(colnames(SG0[, 8:27]))  
 for (i in 1:length(xnam)) {  
 for (j in 1:length(y)) {  
 hÃ¶sh = hopiu[hopiu[, 1] == y[j] & hopiu[, 2] == xnam[i] & hopiu[, 3] == Group,]  
 jeps = SG0  
 r = as.numeric(hÃ¶sh[4][1,])  
 p.valueList = as.numeric(hÃ¶sh[5][1,])  
 rsadj = as.numeric(hÃ¶sh[6][1,])  
 colnames(jeps) = colnames(AllData)[2:dim(AllData)[2]]  
 TreatmentVariables = as.character(hÃ¶sh[2][1,])  
 MediatorVariables = as.character(hÃ¶sh[1][1,])  
 MediatorVariables <- gsub("\\.", "-", MediatorVariables)  
 MediatorVariables <- gsub("X", "", MediatorVariables)  
 if (MediatorVariables == "T-Epi-T") {  
 MediatorVariables[MediatorVariables == "T-Epi-T"] = "T/Epi-T"  
 }  
 Treatment2 = TreatmentVariables  
 colnames(jeps)[colnames(jeps) == 'HOMA-IR'] = 'HOMA.IR'  
 colnames(jeps) <- gsub(" ", "\\.", colnames(jeps))  
 main\_dir <- paste0(c("C://Users//patati//Desktop//Turku//R//", fn, Group, "/"), collapse = "")  
 sub\_dir <- TreatmentVariables  
 if (file.exists(file.path(main\_dir, sub\_dir))) {  
 setwd(file.path(main\_dir, sub\_dir))  
 } else {  
 dir.create(file.path(main\_dir, sub\_dir))  
 setwd(file.path(main\_dir, sub\_dir))  
 }  
 if (MediatorVariables != "T/Epi-T") {  
 jpeg(paste("Correlations plotted", Group, Treatment2, MediatorVariables, ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 } else {  
 jpeg(paste("Correlations plotted\_", Treatment2, 'T.Epi.T', ".jpg"), width = 1000, height = 1000, quality = 100, pointsize = 20, res = 300)  
 }  
 xcx = jeps[, TreatmentVariables]  
 ycy = jeps[, MediatorVariables]  
 vÃ¤h = round(sd(ycy) / 2, 2)  
 BMI\_ordered\_MASLD = ggplot(jeps, aes(y = ycy, x = xcx)) +  
 geom\_point() +  
 xlab(TreatmentVariables) +  
 ylab(MediatorVariables) +  
 geom\_smooth(method = "lm", col = "black") +  
 annotate("text", x = min(xcx), y = max(ycy), hjust = 0, vjust = 0, label = paste0("r = ", round(r, 5))) +  
 annotate("text", x = min(xcx), y = max(ycy) - vÃ¤h, hjust = 0, vjust = 0, label = paste0("p = ", round(p.valueList, 5))) +  
 theme\_classic()  
 print(BMI\_ordered\_MASLD)  
 dev.off()  
 hesh = rbind(hesh, c(y[j], xnam[i], Group, r, p.valueList, rsadj))  
 }  
 }  
   
 # Convert results patientNumbers BMI\_ordered\_MASLD data frame and return  
 hesa = hesh  
 hoi = as.data.frame(hesh)  
 main\_dir <- paste0(c("C://Users//patati//Desktop//Turku//R//", fn), collapse = "")  
 setwd(main\_dir)  
   
 return(list(hopiu))  
}  
  
RunScatterPlotAnalysis=function(AllData,hopiu) {  
 RunAnalysis=c();heijaa=c('All','female','male');   
 fn='metabnorm//covScaled//non\_fdr//'#  
 for (hrt in 1:3) {  
 RunAnalysis=append(RunAnalysis,CreateScatterPlots(AllData,heijaa[hrt],hopiu,aa,bb,fn))}  
 return(RunAnalysis)}  
  
  
# Now I'll be just showing some examples, but this works and if you drive this, you can find all the combos in the folders.  
# E.g.: ".../R/metabnorm/covScaled/non\_fdr/'Subject'/Hexcer/Correlations plotted All Hexcer S .jpg"   
# fyi: do the 'subject' folders (all, male, fem.) separately  
aa=-0.5; bb=0.5; adj='nook'; sig.level=c(.001,0.01, 0.05)  
fn='metabnorm//covScaled//non\_fdr//'#  
metanorm\_S\_non\_fdr=RunAnalysis(CovariatesScaledData,adj,sig.level) # This was already done above  
metanorm\_S\_fdr\_non\_tot=ldply(metanorm\_S\_non\_fdr, data.frame) #Making the above ok for the next:  
non\_fdr\_check=RunScatterPlotAnalysis(CovariatesScaledData,metanorm\_S\_fdr\_non\_tot) # This plots all the combinations, so it takes some time  
  
#The above code in this segment does not get easily better with Copilot, but I let it comment it BMI\_ordered\_MASLD bit.  
  
#Making some 3d plots:  
  
# Load the dataset  
data <- CovariatesScaledData  
  
# Plot BMI\_ordered\_MASLD 3D scatter plot for PFOA, T/Epi-T, and PE  
Create3DScatterPlot(  
 y = data[,'PFOA'],  
 x = data[,'T/Epi-T'],  
 z = data[,'PE'],  
 col = 'black',  
 type = 's',  
 radius = .2,  
 ylab = "PFHxS",  
 xlab = "T/Epi-T",  
 zlab = "PE"  
)  
  
# Fit BMI\_ordered\_MASLD linear model for PFHxS based on T/Epi-T and PE  
my.lm <- lm(data[,'PFHxS'] ~ data[,'T/Epi-T'] + data[,'PE'])  
  
# Extract the variables for the scatter plot  
x <- data[,'T/Epi-T']  
y <- data[,'PFHxS']  
z <- data[,'PE']  
  
# Create BMI\_ordered\_MASLD 3D scatter plot with the fitted plane  
s3d <- scatterplot3d(  
 x, y, z, pch = 20, mar = c(5, 3, 4, 3),  
 main = "3D Scatter Plot of Compounds and Their Residuals",  
 angle = 55, scale.y = 0.5,  
 xlab = "T/Epi-T",  
 ylab = "PFHxS",  
 zlab = "PE"  
)  
s3d$plane3d(my.lm, lty = "dotted")  
  
# Convert coordinates for the original points and the fitted plane  
orig <- s3d$xyz.convert(x, y, z)  
plane <- s3d$xyz.convert(x, y, fitted(my.lm))  
  
# Determine the color and line type for the residuals  
i.negpos <- 1 + (resid(my.lm) > 0)  
segments(orig$x, orig$y, plane$x, plane$y,  
 col = c("blue", "red")[i.negpos], lty = (2:1)[i.negpos])  
  
# Load the dataset again  
data <- CovariatesScaledData  
  
# Plot BMI\_ordered\_MASLD 3D scatter plot for PFOA, AN, and PC\_O  
Create3DScatterPlot(  
 y = data[,'PFOA'],  
 x = data[,'AN'],  
 z = data[,'PC\_O'],  
 col = 'black',  
 type = 's',  
 radius = .2,  
 ylab = "PFOA",  
 xlab = "AN",  
 zlab = "PC\_O"  
)  
  
# Fit BMI\_ordered\_MASLD linear model for PFOA based on AN and PC\_O  
my.lm <- lm(data[,'PFOA'] ~ data[,'AN'] + data[,'PC\_O'])  
  
# Extract the variables for the scatter plot  
y <- data[,'PFOA']  
x <- data[,'AN']  
z <- data[,'PC\_O']  
  
# Create BMI\_ordered\_MASLD 3D scatter plot with the fitted plane  
s3d <- scatterplot3d(  
 x, y, z, pch = 20, mar = c(5, 3, 4, 3),  
 main = "3D Scatter Plot of Compounds and Their Residuals",  
 angle = 55, scale.y = 0.5,  
 xlab = "AN",  
 ylab = "PFOA",  
 zlab = "PC\_O"  
)  
s3d$plane3d(my.lm, lty = "dotted")  
  
# Convert coordinates for the original points and the fitted plane  
orig <- s3d$xyz.convert(x, y, z)  
plane <- s3d$xyz.convert(x, y, fitted(my.lm))  
  
# Determine the color and line type for the residuals  
i.negpos <- 1 + (resid(my.lm) > 0)  
segments(orig$x, orig$y, plane$x, plane$y,  
 col = c("blue", "red")[i.negpos], lty = (2:1)[i.negpos])  
  
# Load the dataset again  
data <- CovariatesScaledData  
  
# Plot BMI\_ordered\_MASLD 3D scatter plot for PFHxA, DHEA, and PC  
Create3DScatterPlot(  
 y = data[,'PFHxA'],  
 x = data[,'DHEA'],  
 z = data[,'PC\_O'],  
 col = 'black',  
 type = 's',  
 radius = .2,  
 ylab = "PFHxA",  
 xlab = "DHEA",  
 zlab = "PC"  
)  
  
# Fit BMI\_ordered\_MASLD linear model for PFHxA based on DHEA and PC  
my.lm <- lm(data[,'PFHxA'] ~ data[,'DHEA'] + data[,'PC'])  
  
# Extract the variables for the scatter plot  
y <- data[,'PFHxA']  
x <- data[,'DHEA']  
z <- data[,'PC']  
  
# Create BMI\_ordered\_MASLD 3D scatter plot with the fitted plane  
s3d <- scatterplot3d(  
 x, y, z, pch = 20, mar = c(5, 3, 4, 3),  
 main = "3D Scatter Plot of Compounds and Their Residuals",  
 angle = 55, scale.y = 0.5,  
 xlab = "DHEA",  
 ylab = "PFHxA",  
 zlab = "PC"  
)  
s3d$plane3d(my.lm, lty = "dotted")  
  
# Convert coordinates for the original points and the fitted plane  
orig <- s3d$xyz.convert(x, y, z)  
plane <- s3d$xyz.convert(x, y, fitted(my.lm))  
  
# Determine the color and line type for the residuals  
i.negpos <- 1 + (resid(my.lm) > 0)  
segments(orig$x, orig$y, plane$x, plane$y,  
 col = c("blue", "red")[i.negpos], lty = (2:1)[i.negpos])  
  
```  
```{r, echo=FALSE, out.width="50%", fig.cap="Correlations Scatter Plotted\_All\_Hexcer S",fig.align="left"}  
knitr::include\_graphics('Correlations plotted All Hexcer S .jpg')  
```  
```{r, echo=FALSE, out.width="50%", fig.cap="Correlations Scatter Plotted\_Female\_Hexcer S",fig.align="left"}  
knitr::include\_graphics('Correlations plotted female Hexcer S .jpg')  
```  
```{r, echo=FALSE, out.width="50%", fig.cap="Correlations Scatter Plotted\_Male\_Hexcer S",fig.align="left"}  
knitr::include\_graphics('Correlations plotted male Hexcer S .jpg')  
```  
```{r, echo=FALSE, out.width="50%", fig.cap="Correlations Scatter Plotted\_All\_C4\_L P5",fig.align="left"}  
knitr::include\_graphics('Correlations plotted All C4\_L P5 .jpg')  
```  
```{r, echo=FALSE, out.width="50%", fig.cap="Correlations Scatter Plotted\_Female\_C4\_L P5",fig.align="left"}  
knitr::include\_graphics('Correlations plotted female C4\_L P5 .jpg')  
```  
```{r, echo=FALSE, out.width="50%", fig.cap="Correlations Scatter Plotted\_Male\_C4\_L P5",fig.align="left"}  
knitr::include\_graphics('Correlations plotted male C4\_L P5 .jpg')  
```  
  
  
# Making Network Plot  
```{r, warning=FALSE,message=FALSE,fig.width=10.0,fig.align="left",results='hide'}  
  
# Following the previous work for doing the network plot (https://r-graph-gallery.com/network.html)  
  
# Here the main data of the plot takes correlations:  
# Remove specific columns from CurrentData  
CurrentData=CurrentData[,!colnames(CurrentData) %in% c('Total\_TG','PFAS','Perfluorodecyl.ethanoic.acid')];   
# Remove columns specified in x4 from CurrentData  
CurrentData=CurrentData[,!colnames(CurrentData) %in% x4]   
# Rename column '17aOH-P4' patientNumbers '17a-OHP4'  
colnames(CurrentData)[colnames(CurrentData)=="17aOH-P4"]="17a-OHP4"   
dat = CurrentData;   
# Remove 'Gender' and 'PatientNumber' columns from dat  
dat=dat[,!colnames(dat) %in% c('Gender','PatientNumber')]   
# Calculate Spearman correlation matrix  
resulta <- (rcorr(as.matrix(dat), type = c('spearman')))$r   
# Set threshold level for connections between vertices (pampulat)  
n\_level=0.9   
# Calculate Nrr and replace NA values with 1  
Nrr=qpNrr(resulta, verbose=FALSE);Nrr[is.na(Nrr)]=1;  
# Create BMI\_ordered\_MASLD condition matrix based on n\_level threshold  
cond=data.frame(as.matrix(Nrr<n\_level))   
# Elementwise matrix multiplication and update column names patientNumbers match row names # https://www.geeksforgeeks.org/elementwise-matrix-multiplication-in-r/  
RN=data.frame(resulta);tes\_t=cond\*RN;tes\_t=as.matrix(tes\_t);resulta=tes\_t;colnames(resulta)=rownames(resulta)   
tes\_t=resulta  
# Calculate lengths of x1 patientNumbers x6 and assign patientNumbers variables BMI\_ordered\_MASLD patientNumbers f  
BMI\_ordered\_MASLD=length(x1)-2;BMI\_ordered\_NAFLD=length(x2);c=length(x3);d=length(x4);e=length(x5);f=length(x6);   
  
# Removing self-correlation  
tes\_t[1:BMI\_ordered\_MASLD,1:BMI\_ordered\_MASLD]=0  
tes\_t[(BMI\_ordered\_MASLD+1):(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD),(BMI\_ordered\_MASLD+1):(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD)]=0  
tes\_t[(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+1):(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+c),(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+1):(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+c)]=0  
tes\_t[(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+c+1):(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+c+e),(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+c+1):(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+c+e)]=0  
tes\_t[(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+c+e+1):(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+c+e+f),(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+c+e+1):(BMI\_ordered\_MASLD+BMI\_ordered\_NAFLD+c+e+f)]=0  
  
# Select BMI\_ordered\_MASLD subset of tes\_t based on column names  
tes\_t = tes\_t[colnames(tes\_t)[7:66], colnames(tes\_t)[7:66]]  
  
# Create BMI\_ordered\_MASLD graph from adjacency matrix  
g <- graph\_from\_adjacency\_matrix(tes\_t, mode="upper", weighted=TRUE, diag=FALSE)  
  
# Convert graph patientNumbers edge list and create BMI\_ordered\_MASLD data frame with weights  
e <- as\_edgelist(g)  
df <- as.data.frame(cbind(e, E(g)$weight))  
  
# Convert the third column patientNumbers numeric  
df[, 3] = as.numeric(df[, 3])  
hoi=df   
hoi=hoi[!duplicated(hoi[,c(1,2)]),] # Remove duplicate rows based on the first two columns  
  
# Load igraph library if needed  
# https://r-graph-gallery.com/249-igraph-network-map-BMI\_ordered\_MASLD-color.html  
# Create BMI\_ordered\_MASLD data frame for links with source, target, and importance  
links <- data.frame(  
 source=c("A","A", "A", "A", "A","J", "B", "B", "C", "C", "D","I"),  
 target=c("B","B", "C", "D", "J","A","E", "F", "G", "H", "I","I"),  
 importance=(sample(1:4, 12, replace=T)))   
# Set column names of hoi patientNumbers match links  
colnames(hoi)=colnames(links)   
links=hoi  
# Get distinct sources and rename column patientNumbers 'label'  
sources=hoi %>% distinct(source) %>% rename(source='label')  
# Get distinct targets and rename column patientNumbers 'label'  
destinations=hoi %>% distinct(target) %>% rename(target ='label')   
# Merge sources and destinations into nodess  
nodess <- full\_join(sources, destinations, by = "label")   
  
# The names of the variable steroidGroups, such as xc for contaminants  
xc=x5[x5 %in% nodess$label]  
xb=x3[x3 %in% nodess$label]  
xl=x6[x6 %in% nodess$label]  
x2[x2 =='17aOH-P4']='17a-OHP4' #Next time check these wrong names early on... :)  
xs=x2[x2 %in% nodess$label]  
nodess$label=c(xc,xb,xl,xs)  
  
# Create BMI\_ordered\_MASLD data frame 'nodes' with two columns: 'name' and 'carac'  
# 'name' is taken from the first column of 'nodess'  
# 'carac' is BMI\_ordered\_MASLD categorical variable indicating the type of each node  
nodes <- data.frame(name=nodess[,1],   
 carac=( c(rep("Contaminants",length(xc)),rep("Bile Acids",length(xb)),rep("Lipids",length(xl)),rep("Steroids",length(xs))))) #range on kaikki +1  
  
# Convert the data frame 'links' and 'nodes' into an igraph object  
network <- graph\_from\_data\_frame(d=links, vertices=nodes, directed=F)   
  
# Load necessary libraries for color palettes and graphics (library(RColorBrewer), library(ragg)) if needed  
  
# Set the font patientNumbers 'Calibri (Body)'  
windowsFonts(A = windowsFont("Calibri (Body)"))  
  
# Define BMI\_ordered\_MASLD palette of 4 colors  
coul <- c('#B2BEB5','Green','Red','Orange')   
  
# Create BMI\_ordered\_MASLD vector of colors corresponding patientNumbers the 'carac' variable in the network  
my\_color <- coul[as.numeric(as.factor(V(network)$carac))]  
  
# Save the plot as BMI\_ordered\_MASLD JPEG file with specified dimensions and quality  
jpeg('network.jpg', width=4, height=4.7, units="in", quality=100, pointsize=7, res=1000)  
  
# Plot the network with specified parameters  
plot(network, mode = "circle", vertex.color=my\_color, vertex.size = 10,  
 edge.arrow.size = 0.8, vertex.label.cex = 0.35, edge.width=as.numeric(E(network)$importance)\*6.00 )  
  
# Add BMI\_ordered\_MASLD legend patientNumbers the plot  
legend("topright", legend=levels(as.factor(V(network)$carac)),  
 col = coul, bty = "n", pch=20, pt.cex = 1.3, cex = 1.3, text.col=coul,  
 horiz = FALSE, inset = c(0.65, 0.8))  
  
# Close the JPEG device  
dev.off()  
  
# Read the saved JPEG image  
eoh='network.jpg'  
my\_image <- image\_read(eoh)  
  
# Convert the image patientNumbers SVG format and save it  
my\_svg <- image\_convert(my\_image, format="svg")  
image\_write(my\_svg, paste(eoh,".svg"))  
  
# Display the figure and convert it patientNumbers PDF  
knitr::include\_graphics(eoh)  
daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
  
# Copilot helped here with the commenting.  
  
  
```  
  
# Making Causal Mediation Analysis  
```{r, warning=FALSE,message=FALSE,fig.align="left"}  
  
# The basic hypothesis. All are variables (y~x+m;m~x)  
RunMediationAnalysis = function(TreatmentVariables, MediatorVariables, OutcomeVariables, AllData, Group, name, simss, t.valueList, test, sick, sick\_group) {  
 # Determine the condition based on the group  
 if (Group == 'female') {  
 cond = AllData[,'Gender'] == min(AllData[,'Gender'])  
 } else if (Group == 'male') {  
 cond = AllData[,'Gender'] == max(AllData[,'Gender'])  
 } else if (Group == 'All') {  
 cond = rep(TRUE, dim(AllData)[1])  
 }  
   
 # Filter the dataset based on the condition and sickness status  
 tv\_red = c()  
 if (sick == 'yes') {  
 tv\_red = AllData[cond & as.vector(sick\_group),]  
 } else {  
 tv\_red = AllData[cond,]  
 }  
   
 # Extract the treatment, mediator, and outcome variables  
 X <- tv\_red[, TreatmentVariables] # Standard values did not give errors  
 M <- tv\_red[, MediatorVariables]  
 Y <- tv\_red[, OutcomeVariables] # "Steatosis.Grade.0.To.3", "Fibrosis.Stage.0.patientNumbers.4", "Necroinflammation", "HOMA-IR"  
   
 # Combine the variables into BMI\_ordered\_MASLD single data frame  
 Data <- cbind(X, M, Y)  
 colnames(Data) <- gsub(" ", "\_", colnames(Data))  
 Data = data.frame(Data)  
   
 # Define control and treatment values for the mediation analysis  
 control.value = colMins(as.matrix(X)) # Test also with colMedians  
 treat.value = colMaxs(as.matrix(X))  
   
 # Initialize vectors for storing variable names  
 x = c()  
 m = c()  
 y = c()  
   
 # Create variable names for the treatment, mediator, and outcome  
 for (i in 1:length(colnames(X))) {  
 x = append(x, paste("Data[, ", i , "]", sep = ""))  
 }  
 for (j in (dim(X)[2] + 1):(length(colnames(M)) + dim(X)[2])) {  
 m = append(m, paste("Data[, ", j , "]", sep = ""))  
 }  
 for (z in (dim(M)[2] + dim(X)[2] + 1):(dim(Data)[2])) {  
 y = append(y, paste("Data[, ", z , "]", sep = ""))  
 }  
   
 # Initialize vectors for storing results  
 med\_out = c()  
 res = c()  
 tmp = c()  
 rn = c()  
 i=1;j=1;z=1  
   
 # Loop through each combination of outcome, mediator, and treatment  
 for (i in 1:length(y)) {  
 for (j in 1:length(m)) {  
 for (z in 1:length(x)) {  
 # Define the formula for the mediator model (M ~ X)  
 fmla1 <- as.formula(paste(paste(m[j], collapse = "+"), " ~ ", paste(x[z], collapse = "+")))  
 BMI\_ordered\_NAFLD = lm(fmla1, Data)  
   
 # Define the formula for the outcome model (Y ~ M + X)  
 xm = paste(paste(c(x[z], m[j]), collapse = "+"))  
 fmla2 <- as.formula(paste(y[i], " ~ ", xm))  
 c = lm(fmla2, Data)  
   
 # Perform the mediation analysis  
 if (t.valueList == 'no') {  
 med\_oute = mediation::mediate(BMI\_ordered\_NAFLD, c, treat = x[z], mediator = m[j], sims = simss)  
 } else if (t.valueList == 'yes') {  
 med\_oute = mediation::mediate(BMI\_ordered\_NAFLD, c, treat = x[z], mediator = m[j], sims = simss, control.value = control.value[z], treat.value = X[test, z])  
 } else if (t.valueList == 'minmax') {  
 med\_oute = mediation::mediate(BMI\_ordered\_NAFLD, c, treat = x[z], mediator = m[j], sims = simss, control.value = control.value[z], treat.value = treat.value[z])  
 }  
   
 # Summarize the mediation analysis results  
 med\_out = summary(med\_oute)  
 tmp = c(med\_out$d0, med\_out$d0.p, med\_out$d0.ci[1], med\_out$d0.ci[2], med\_out$z0, med\_out$z0.p, med\_out$z0.ci[1], med\_out$z0.ci[2], med\_out$n1, med\_out$n1.p, med\_out$n1.ci[1], med\_out$n1.ci[2], med\_out$tau.coef, med\_out$tau.p, med\_out$tau.ci[1], med\_out$tau.ci[2])  
 res <- rbind(res, tmp)  
 rn = append(rn, paste(colnames(X)[z], colnames(M)[j], colnames(Y)[i], sep = " "))  
 remove(tmp)  
 }  
 }  
 }  
   
 # Assign row and column names patientNumbers the results  
 rownames(res) = rn  
 colnames(res) = c('ACME', 'd0.p', 'd0.ci\_l', 'd0.ci\_u', 'ADE', 'z0.p', 'z0.ci\_l', 'z0.ci\_u', 'Proportion Mediated', 'n1.p', 'n.ci\_l', 'n1.ci\_u', 'Total Effect', 'tau.p', 'tau.ci\_l', 'tau.ci\_u')  
   
 # Order the results by p-value  
 res = res[order(res[, 2]),]  
   
 # Clean up row names  
 rownames(res) <- gsub("X11", "11", rownames(res))  
 rownames(res) <- gsub("X17", "17", rownames(res))  
   
 # Write the results patientNumbers an Excel file  
 write.xlsx(res, file = paste(name, Group, date, '.xlsx'), append = FALSE, row.names = TRUE)  
   
 return(res)  
}  
  
# Testing hypothesis one:  
RunEssentialAnalysis = function(TreatmentVariables, MediatorVariables, OutcomeVariables, CombinedData, Group, name, simss, t.valueList, test, sick, sick\_group, fn, lkm, date, joo, ip) {  
 # Set the working directory patientNumbers the specified path  
 hoi1 = paste("C:/Users/patati/Desktop/Turku/R/basic causal mediation with the counterfactuals/", fn, sep = '')  
 setwd(hoi1) # Set the working directory  
   
 # Define the name for the output files  
 name = paste(simss, 'basic\_divisions')  
   
 # Perform mediation analysis for all steroidGroups  
 Group = 'All'  
 uh7ma = RunMediationAnalysis(TreatmentVariables, MediatorVariables, OutcomeVariables, AllData, Group, name, simss, t.valueList, test, sick, sick\_group)  
 try({uh7ma}, {uh7ma = data.frame(0)}) # Handle errors gracefully  
 save(uh7ma, file = paste(fn, 'all.RData')) # Save the results  
   
 # Perform mediation analysis for female group  
 Group = 'female'  
 uh7f = RunMediationAnalysis(TreatmentVariables, MediatorVariables, OutcomeVariables, AllData, Group, name, simss, t.valueList, test, sick, sick\_group)  
 try({uh7f}, {uh7f = data.frame(0)}) # Handle errors gracefully  
 save(uh7f, file = paste(fn, 'female.RData')) # Save the results  
   
 # Perform mediation analysis for male group  
 Group = 'male'  
 uh7m = RunMediationAnalysis(TreatmentVariables, MediatorVariables, OutcomeVariables, AllData, Group, name, simss, t.valueList, test, sick, sick\_group)  
 try({uh7m}, {uh7m = data.frame(0)}) # Handle errors gracefully  
 save(uh7m, file = paste(fn, 'male.RData')) # Save the results  
}  
  
# Set the number of simulations and other parameters  
simss = 100  
name = 'Jaot\_OK\_basica'  
joo = 'ei'  
ip = 1  
  
# Extract specific columns from the dataset for analysis  
ccova = CombinedData[, c("Steatosis.Grade.0.To.3", "Fibrosis.Stage.0.patientNumbers.4", "Necroinflammation", "HOMA-IR")]  
  
# Define BMI\_ordered\_MASLD group based on BMI\_ordered\_MASLD condition (sum of specific columns > 4)  
sick\_group = rowSums(ccova) > 4  
  
# Define file names for different outcomes  
file\_names = c("Steatosis", "Fibrosis", "Necroinflammation", "HOMAIR", 'Menopause')  
  
# Set additional parameters  
lkm = 30  
joo = 'ei'  
ip = 1  
sick = 'yes'  
t.valueList = 'no'  
Group = 'All'  
# date='houdeesh'  
  
# Set parameters for the 'All' group analysis  
joo = 'joo'  
sick = 'no'  
  
# Extract specific columns from the dataset for analysis  
ccova = CombinedData[, c("Steatosis.Grade.0.To.3", "Fibrosis.Stage.0.patientNumbers.4", "Necroinflammation", "HOMA-IR")]  
  
# Define the file name and group based on BMI\_ordered\_MASLD condition (sum of specific columns > 4)  
fn = 'All'  
sick\_group = rowSums(ccova) > 4  
  
# Driving the analysis takes more than 1h (with BMI\_ordered\_MASLD basic laptop 2023,100sims and all three steroidGroups (all, female and male)) so drive the below only if needed:  
# RunEssentialAnalysis(TreatmentVariables, MediatorVariables, OutcomeVariables,AllData,Group,name,simss,t.valueList,test,sick, sick\_group,fn,lkm,date,joo,ip)  
# setwd("C:/Users/patati/Desktop/Turku/R/basic causal mediation with the counterfactuals/") #check this if needed...  
  
# Error in mediate(BMI\_ordered\_NAFLD, c, treat = x[z], mediator = m[j], sims = simss) :   
 # unused arguments (treat = x[z], mediator = m[j], sims = simss) ...  
  
# In case, the 'case' sample/subject analysis would be needed:   
# Steatosis  
# ccovae=CombinedData[,c("Steatosis.Grade.0.To.3")]; sick\_group=ccovae>0 #toth# # hist(ccovae,breaks=100) # hist(ccova[,'HOMA-IR'],breaks=100)  
# fn=file\_names[1];  
# RunEssentialAnalysis(TreatmentVariables, MediatorVariables, OutcomeVariables,AllData,Group,name,simss,t.valueList,test,sick,sick\_group,fn,lkm,date,joo,ip)  
# #Fibrosis  
# ccovae=CombinedData[,c("Fibrosis.Stage.0.patientNumbers.4")];   
# sick\_group=ccovae>0 #toth#  
# fn=file\_names[2];   
# RunEssentialAnalysis(TreatmentVariables, MediatorVariables, OutcomeVariables,AllData,Group,name,simss,t.valueList,test,sick,sick\_group,fn,lkm,date,joo,ip)  
# #Necroinfl.  
# ccovae=CombinedData[,c("Necroinflammation")]; sick\_group=ccovae>0 #toth#  
# fn=file\_names[3];  
# RunEssentialAnalysis(TreatmentVariables, MediatorVariables, OutcomeVariables,AllData,Group,name,simss,t.valueList,test,sick,sick\_group,fn,lkm,date,joo,ip)  
# Homa # remember patientNumbers always test the function with minimun number setting or as light parameters as possible patientNumbers get it through... before big runs  
# joo='ei';sick='yes'  
# ccovae=CombinedData[,c("HOMA-IR")]; sick\_group=ccovae>1.5 #toth#  
# fn=file\_names[4];   
# RunEssentialAnalysis(TreatmentVariables, MediatorVariables, OutcomeVariables,AllData,Group,name,simss,t.valueList,test,sick,sick\_group,fn,lkm,date,joo,ip)  
  
# In addition, then there is also possibility for elaborations, i.e. the adjustments, of the models,   
# but I'll leave it from here for the time being... (please do ask if needed: patati 'tilde' utu dot fi) :)  
# fyi: https://bookdown.org/content/b472c7b3-ede5-40f0-9677-75c3704c7e5c/more-than-one-mediator.html  
  
  
  
#Some commenting with Copilot here.  
  
  
```  
  
  
  
  
  
  
  
  
  
# Making Heatmaps for Indirect and Direct Effects  
```{r, warning=FALSE,message=FALSE,fig.width=10.0,fig.align="left"}  
# setwd("C:/Users/patati/Desktop/Turku/R/tests6/tests\_basic/") #check this if needed...  
library(readxl)  
all\_all=read\_xlsx(path = "C:/Users/patati/Desktop/Turku/R/tests6/tests\_basic/100basic All tikka3624 .xlsx") # #\_1  
# all\_all=read\_xlsx(path = "C:/Users/patati/Desktop/Turku/R/hypo\_basic/100 hypo\_b\_no\_not sick All tikka221024 .xlsx") # #\_2 :)  
all\_all=as.data.frame(all\_all); all\_all=all\_all[!is.na(all\_all[,1]),];rownames(all\_all)=all\_all[,1]; all\_all=all\_all[,2:dim(all\_all)[2]]; all\_all=all\_all[rev(order(all\_all[,1])),]  
all\_all=all\_all; #all\_all=all\_all[all\_all[,1]>0,]  
#https://stats.stackexchange.com/questions/282155/causal-mediation-analysis-negative-indirect-and-total-effect-positive-direct  
#https://www.researchgate.net/post/How\_can\_I\_interpret\_a\_negative\_indirect\_effect\_for\_significant\_mediation  
#https://stackoverflow.com/questions/31518150/gsub-in-r-is-not-replacing-dot replacing dot  
steroidGroups=groupValues  
steroidGroups[,'Abbreviation'][steroidGroups[,'Abbreviation']=='17aOH-P4']='17a-OHP4'  
  
#Switch = 0: PFAS vs steroids; switch=1: PFAS vs BAs and lipids, switch=2: steroids vs BAs and lipids (0-2 with both ACME and ADE (z='dir'))  
  
CreateHeatmaps=function(hoi, rt2, switch, mn, z, corr, date, neg) {  
 indir=c(); dir=c(); ip=c(); rn=c(); rn2=c()  
 OutcomeVariables=colnames(CovariatesNonScaledData)[c(29:51,59:71)]; # The final dataframe is shorter or the like so there were less variables here...  
 TreatmentVariables=colnames(CovariatesNonScaledData)[52:58];  
 ## https://sparkbyexamples.com/r-programming/r-remove-from-vector-with-examples/  
   
 # Direct mediation analysis  
 if (switch==1) {  
 Mediator\_ok=OutcomeVariables[OutcomeVariables %in% names(table(hoi[1:dim(hoi)[1],c(3)]))]  
 for (i in 1:7) {  
 for (j in 1:length(Mediator\_ok)) {  
 if (z=='dir') {  
 ap=rt2[which(hoi[,1]==TreatmentVariables[i] & hoi[,3]==Mediator\_ok[j]),];   
 if (!is.na(median(ap[,'ADE']))) {  
 if (median(ap[,'ADE']) < quantile(rt2[,'ADE'],0.5)) {  
 apa=min(ap[,'ADE']);  
 ape=ap[ap[,'ADE']==min(ap[,'ADE']),'z0.p']  
 } else {  
 apa=max(ap[,'ADE']);  
 ape=ap[ap[,'ADE']==max(ap[,'ADE']),'z0.p']  
 }  
 } else {  
 apa=0; ape=1  
 }  
 indir=append(indir,apa) # or c(1) hoi 1 or 5 (5 is orig)  
 ip=append(ip,ape)  
 } else {  
 ap=rt2[which(hoi[,1]==TreatmentVariables[i] & hoi[,3]==Mediator\_ok[j]),];   
 if (!is.na(median(ap[,'ACME']))) {  
 if (median(ap[,'ACME']) < quantile(rt2[,'ACME'],0.5)) {  
 apa=min(ap[,'ACME']);  
 ape=ap[ap[,'ACME']==min(ap[,'ACME']),'d0.p']  
 } else {  
 apa=max(ap[,'ACME']);  
 ape=ap[ap[,'ACME']==max(ap[,'ACME']),'d0.p']  
 }  
 } else {  
 apa=0; ape=1  
 }  
 indir=append(indir,apa) # or c(1) hoi 1 or 5 (5 is orig)  
 ip=append(ip,ape)  
 }  
 rn=append(rn,hoi[,3][which(hoi[,1]==TreatmentVariables[i] & hoi[,3]==Mediator\_ok[j])[1]]) # change this...  
 rn2=append(rn2,hoi[,1][which(hoi[,1]==TreatmentVariables[i] & hoi[,3]==Mediator\_ok[j])[1]])  
   
 Matrix <- matrix(0, nrow = length(TreatmentVariables), ncol = length(Mediator\_ok))  
 myData <- data.frame(matrix = Matrix)  
 colnames(myData) <- Mediator\_ok; rownames(myData) <- TreatmentVariables  
 }  
 }  
 } else if (switch==0) {  
 # Indirect mediation analysis  
 Mediator\_ok=colnames(AllData)[9:28][colnames(AllData)[9:28] %in% names(table(hoi[1:dim(hoi)[1],c(2)]))]  
 for (i in 1:7) {  
 for (j in 1:length(Mediator\_ok)) {  
 if (z=='dir') {  
 ap=rt2[which(hoi[,1]==TreatmentVariables[i] & hoi[,2]==Mediator\_ok[j]),];   
 if (!is.na(median(ap[,'ADE']))) {  
 if (median(ap[,'ADE']) < quantile(rt2[,'ADE'],0.5)) {  
 apa=min(ap[,'ADE']);  
 ape=ap[ap[,'ADE']==min(ap[,'ADE']),'z0.p']  
 } else {  
 apa=max(ap[,'ADE']);  
 ape=ap[ap[,'ADE']==max(ap[,'ADE']),'z0.p']  
 }  
 } else {  
 apa=0; ape=1  
 }  
 indir=append(indir,apa) # or c(1) hoi 1 or 5 (5 is orig)  
 ip=append(ip,ape)  
 } else {  
 ap=rt2[which(hoi[,1]==TreatmentVariables[i] & hoi[,2]==Mediator\_ok[j]),];   
 if (!is.na(median(ap[,'ACME']))) {  
 if (quantile(ap[,'ACME'],0.50) < quantile(rt2[,'ACME'],0.5)) {  
 apa=min(ap[,'ACME']);  
 ape=ap[ap[,'ACME']==min(ap[,'ACME']),'d0.p']  
 } else {  
 apa=max(ap[,'ACME']);  
 ape=ap[ap[,'ACME']==max(ap[,'ACME']),'d0.p']  
 }  
 } else {  
 apa=0; ape=1  
 }  
 indir=append(indir,apa) # or c(1) hoi 1 or 5 (5 is orig)  
 ip=append(ip,ape)  
 }  
 rn=append(rn,hoi[,2][which(hoi[,1]==TreatmentVariables[i] & hoi[,2]==Mediator\_ok[j])[1]]) # change this...  
 rn2=append(rn2,hoi[,1][which(hoi[,1]==TreatmentVariables[i] & hoi[,2]==Mediator\_ok[j])[1]])  
   
 Matrix <- matrix(0, nrow = length(TreatmentVariables), ncol = length(Mediator\_ok))  
 myData <- data.frame(matrix = Matrix)  
 colnames(myData) <- Mediator\_ok; rownames(myData) <- TreatmentVariables  
 }  
 }  
 } else if (switch==2) {  
 TreatmentVariables=colnames(AllData)[9:28]; # These names are BMI\_ordered\_MASLD bit mixed, but the idea is ok.  
 Mediator\_ok=OutcomeVariables[OutcomeVariables %in% names(table(hoi[1:dim(hoi)[1],c(3)]))]  
   
 for (i in 1:length(TreatmentVariables)) {  
 for (j in 1:length(Mediator\_ok)) {  
 if (z=='dir') {  
 ap=rt2[which(hoi[,2]==TreatmentVariables[i] & hoi[,3]==Mediator\_ok[j]),];   
 if (!is.na(median(ap[,'ADE']))) {  
 if (median(ap[,'ADE']) < quantile(rt2[,'ADE'],0.5)) {  
 apa=min(ap[,'ADE']);  
 ape=ap[ap[,'ADE']==min(ap[,'ADE']),'z0.p']  
 } else {  
 apa=max(ap[,'ADE']);  
 ape=ap[ap[,'ADE']==max(ap[,'ADE']),'z0.p']  
 }  
 } else {  
 apa=0; ape=1  
 }  
 indir=append(indir,apa) # or c(1) hoi 1 or 5 (5 is orig)  
 ip=append(ip,ape)  
 } else {  
 ap=rt2[which(hoi[,2]==TreatmentVariables[i] & hoi[,3]==Mediator\_ok[j]),];   
 if (!is.na(median(ap[,'ACME']))) {  
 if (median(ap[,'ACME']) < quantile(rt2[,'ACME'],0.5)) {  
 apa=min(ap[,'ACME']);  
 ape=ap[ap[,'ACME']==min(ap[,'ACME']),'d0.p']  
 } else {  
 apa=max(ap[,'ACME']);  
 ape=ap[ap[,'ACME']==max(ap[,'ACME']),'d0.p']  
 }  
 } else {  
 apa=0; ape=1  
 }  
 indir=append(indir,apa) # or c(1) hoi 1 or 5 (5 is orig)  
 ip=append(ip,ape)  
 }  
 rn=append(rn,hoi[,3][which(hoi[,2]==TreatmentVariables[i] & hoi[,3]==Mediator\_ok[j])[1]]) # change this...  
 rn2=append(rn2,hoi[,2][which(hoi[,2]==TreatmentVariables[i] & hoi[,3]==Mediator\_ok[j])[1]])  
 Matrix <- matrix(0, nrow = length(TreatmentVariables), ncol = length(Mediator\_ok))  
 myData <- data.frame(matrix = Matrix)  
 colnames(myData) <- Mediator\_ok; rownames(myData) <- TreatmentVariables  
 }  
 }  
 } # You need three of these, yes :) ;print(ap[,'ADE']) print(rownames(ap[ap[,'ADE']==min(ap[,'ADE']),]))  
   
 tot=cbind(rn2,rn,indir) # or indir or dir  
 tot=tot[!is.na(tot[,1]),]  
 tot=as.data.frame(tot)  
  
   
 # Here you would need patientNumbers have an evaluation for the 'else', if it is negative, it needs patientNumbers be max negative, and vice versa for the  
 # There does not seem patientNumbers be doubles: table(tot[,1])  
   
 tot[,3]=as.numeric(tot[,3])  
   
 library(reshape2)  
 jops=dcast(tot, rn2~rn, value.var='indir')  
 jops[is.na(jops)]=0  
 rownames(jops)=jops[,1]  
 jops=jops[,2:dim(jops)[2]]  
 jops=as.data.frame(jops)  
 jopsr=matrix(as.numeric(unlist(jops)),nrow=dim(jops)[1],ncol=dim(jops)[2])  
 colnames(jopsr)=colnames(jops); rownames(jopsr)=rownames(jops)  
   
 # Ensure all rows and columns from myData are present in jopsr  
 if (sum(!rownames(myData) %in% rownames(jopsr)) > 0) {  
 to\_df=rownames(myData)[!rownames(myData) %in% rownames(jopsr)]  
 jopsr=rbind(jopsr, myData[to\_df,]); jopsr=jopsr[rownames(myData),]  
 }  
 if (sum(!colnames(myData) %in% colnames(jopsr)) > 0) {  
 to\_df=colnames(myData)[!colnames(myData) %in% colnames(jopsr)]  
 jopsr=cbind(jopsr, myData[,to\_df]); jopsr=jopsr[,colnames(myData)]  
 }  
   
 tot=cbind(rn2, rn, ip)  
 tot=tot[!is.na(tot[,1]),]  
 tot=as.data.frame(tot)  
 tot[,3]=as.numeric(tot[,3])  
   
 jopsa=dcast(tot, rn2~rn, value.var='ip')  
 jopsa[is.na(jopsa)]=0  
 rownames(jopsa)=jopsa[,1]  
 jopsa=jopsa[,2:dim(jopsa)[2]]  
 jopsra=matrix(as.numeric(unlist(jopsa)),nrow=dim(jopsa)[1],ncol=dim(jopsa)[2])  
 colnames(jopsra)=colnames(jopsa); rownames(jopsra)=rownames(jopsa)  
   
 if (sum(!rownames(myData) %in% rownames(jopsra)) > 0) {  
 to\_df=rownames(myData)[!rownames(myData) %in% rownames(jopsra)]  
 jopsra=rbind(jopsra, myData[to\_df,]); jopsra=jopsra[rownames(myData),]  
 }  
 if (sum(!colnames(myData) %in% colnames(jopsra)) > 0) {  
 to\_df=colnames(myData)[!colnames(myData) %in% colnames(jopsra)]  
 jopsra=cbind(jopsra, myData[,to\_df]); jopsra=jopsra[,colnames(myData)]  
 }  
   
 if (switch==1) {  
 # For direct effects  
 } else if (switch==0) {  
 # For indirect effects  
 jopsra=jopsra[,steroidGroups[,'Abbreviation'][steroidGroups[,'Abbreviation'] %in% colnames(jopsra)]]  
 jopsr=jopsr[,steroidGroups[,'Abbreviation'][steroidGroups[,'Abbreviation'] %in% colnames(jopsr)]]  
 } else if (switch==2) {  
 jopsra=jopsra[steroidGroups[,'Abbreviation'],]  
 jopsr=jopsr[steroidGroups[,'Abbreviation'],]  
 }  
   
 # With the neg. you do not put these:  
 if (dim(resulta1)[2]==36) {  
 OutcomeVariables=colnames(CovariatesNonScaledData)[c(29:51,59:71)];  
 OutcomeVariables=OutcomeVariables[c(1,23,2:22,24:length(OutcomeVariables))]  
 resulta1=resulta1[,OutcomeVariables[OutcomeVariables %in% colnames(resulta1) ]];p.mat.a1=p.mat.a1[,OutcomeVariables[OutcomeVariables %in% colnames(p.mat.a1) ]] }  
 for (i in 1:dim(resulta1)[1]) {for (j in 1:dim(resulta1)[2]) {if (resulta1[i,j]==0) {p.mat.a1[i,j]=0.5}}}   
 # resulta1 <- t(resulta1);  
 # p.mat.a1 <- t(p.mat.a1)  
 # ou=round(min(c(abs(max(resulta1)),abs(min(resulta1)))),2)  
 # op=ou-0.01  
 heps=abs(round((min(resulta1)),1)); hepsa=abs(round((max(resulta1)),1))  
 heps2=abs(round((max(resulta1)-0.01),3));heps3=abs(round((min(resulta1)+0.01),3));  
   
 if (hepsa-heps >=0 ) {resulta1[resulta1 >= heps2] = hepsa; resulta1[resulta1 <= -heps3] = -hepsa; heps=hepsa} else   
 if (hepsa-heps < 0 ) {resulta1[resulta1 <= -heps] = -heps; resulta1[resulta1 >= heps2] = heps}   
   
 path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; setwd(path) #check this if needed...  
 jpeg(paste("Heatmap of high",date, mn,neg,".jpg"), width = width, height = height, quality = 100,pointsize = 14, res=300);# par( ps=ps)# par(cex.lab=90) 22 18  
 # col = brewer.pal(n = 9, name = "YlOrRd")  
 order="original"; range='orig';corre='no\_renormaa'; type='full'; method='color';ga='All';gf='Female';gm='Male' #color square  
 cl.offset=20;cl.length=7;cl.cex = 1.45;pch.cex=2.45;pch=2;cl.pos = 'r'; #cl.offset=2;cl.length=5;cl.cex = 1.3;pch.cex=1.95;pch=14;  
   
 col=colorRampPalette(c('blue', 'white','orange'), alpha = TRUE)(150)  
 if (neg=='no') {col=colorRampPalette(c( 'white','orange'), alpha = TRUE)(150)} else if (neg=='yes')  
 {col=colorRampPalette(c('blue', 'white'), alpha = TRUE)(150)} else {col=col}  
   
 # if (corr==TRUE) {if (min(as.matrix(resulta1))< -1 | max(as.matrix(resulta1))> 1) {resulta1=rango(resulta1,-1,1)}} else if (min(as.matrix(resulta1)) >= 0) {resulta1=rango(resulta1,-1,1)} #  
 # resulta1=rango(resulta1,-1,1)  
 # if (min(as.matrix(resulta1)) >= 0 | max(as.matrix(resulta1)) <= 0) {resulta1=rango(resulta1,-1,1)}  
   
 corrplot(as.matrix(resulta1), type = type, order = order,method=method, p.mat=as.matrix(p.mat.a1), tl.col = "black",   
 cl.cex = cl.cex, pch.cex=pch.cex, pch.col='black',pch=pch,#pitikÃ¶ vain pch lisÃ¤tÃ¤ pch vÃ¤riin vÃ¤riin..'#FEE12B'  
 sig.level = c(0.05),cl.pos = cl.pos, insig = "label\_sig", cl.offset=cl.offset,cl.length=cl.length, #.001, .05, .2  
 tl.srt = 90, diag = TRUE,col=col,is.corr = corr,col.lim=c(-heps,heps))   
 #only in age...0.001,col.lim=c(-heps,heps) #rev(COL2('RdBu')[25:(length(COL2('RdBu'))-25)]) col.lim=c(-1,1) col.lim=c(-heps,heps)  
 #non were significant in neg... but after mody yes!  
   
 dev.off(); eoh=paste("Heatmap of high",date, mn,neg,".jpg"); daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
 my\_image <- image\_read(eoh);my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
   
 return(list(resulta1, p.mat.a1))  
}  
  
# Switch = 0: PFAS vs steroids;   
# Switch = 1: PFAS vs BAs and lipids,   
# Switch = 2: steroids vs BAs and lipids (0-2 with both ACME and ADE (z='dir'))  
# corr=TRUE;z='idir' # uliulie2=CreateHeatmaps(hoi, switch=1,mn='indiruush',z,corr,date); dim(uliulie2[[1]])   
  
# #Let's get this comparable done:  
# First the 'matrisse'  
corr=FALSE; z='dir'; switch=2   
u3=all\_all; c1=c(); # mn='basicas' #  
c1= u3 #[u3[,'ADE'] < ADEMedian & DV<ADEVar,] #& u3[,'z0.p']<ADEpval  
rt2=c1[complete.cases(c1), ] # 0.49 is optimal p value cutoff patientNumbers get dim(mat[[2]])[2] as 36  
hoi = c(); hoi=scan(text=rownames(rt2), what="") # scan(text=rownames(rt2), what="")  
hoi = matrix(hoi, ncol = 3, byrow = TRUE); colnames(hoi)=c('Contaminants','Steroids','Bile Acids or Lipids') # ,'Gender') ,'Desig.')  
hoi[,c(2)] <- gsub("\\.", "-", hoi[,c(2)] ); hoi[,'Steroids' ][hoi[,'Steroids' ]=='17aOH-P4']='17a-OHP4'  
mn=paste0(z,'ade\_s\_vs\_bal\_no order3b'); neg='else'  
mat=CreateHeatmaps(hoi, rt2, switch, mn, z, corr, date, neg); # dim(mat[[2]])[2] # Indirect effect, kasvata... dim(mat[[2]])[2] == 36  
  
corr=FALSE; z='dir'; switch=1;   
u3=all\_all;  
c1=c() #  
c1= u3 #[u3[,'ADE'] < ADEMedian & DV<ADEVar,] #& u3[,'z0.p']<ADEpval  
c1=c1[rev(order(c1[,'ADE'])),]; #  
c1=c1[c1[,'z0.p'] < 0.89, ] # 0.98/0.81/0.45/0.53/0.1... but gives 33 columns so use the other then..  
# c1=c1[c1[,'d0.p'] < 0.50, ]  
# let us start with the above optima... 643/or similar is dim(c1)[1] so need higher p patientNumbers get all; (dim(c1)[1]-1)  
# Check if this needs patientNumbers be ADE instead...  
c3=c1[c1[,'ADE'] < 0,] # -0.01,]#quantile(c1[,'ACME'])[2]  
c1=c3;  
rt2=c1[complete.cases(c1), ]  
hoi=c(); hoi=scan(text=rownames(rt2), what="") # scan(text=rownames(rt2), what="")  
hoi=matrix(hoi, ncol = 3, byrow = TRUE); colnames(hoi)=c('Contaminants','Steroids','Bile Acids or Lipids') # ,'Gender') ,'Desig.')  
hoi[,c(2)] <- gsub("\\.", "-", hoi[,c(2)] ); hoi[,'Steroids' ][hoi[,'Steroids' ]=='17aOH-P4']='17a-OHP4'  
# mat\_neg=list(c(1,2),c(3,4))  
mn=paste0(z,'neg'); neg='yes'  
mat\_neg=CreateHeatmaps(hoi, rt2, switch, mn, z, corr, date, neg);   
# dim(mat\_neg[[2]])[2] # kasvata n, jotta dim(mat\_pos[[2]])[2] yhtÃ¤kuin kuin length(c(x3,x6)), i.e. 36  
  
u3=all\_all; c1=c(); # mn='posae'  
c1= u3 #[u3[,'ADE'] < ADEMedian & DV<ADEVar,] #& u3[,'z0.p']<ADEpval  
c1=c1[rev(order(c1[,'ACME'])),]; # Check acmes and ades  
c1=c1[c1[,'z0.p']<0.89, ]  
c2=c1[c1[,'ADE'] >= 0,]  
c1=c2; #   
# c1= c1[sample(1:nrow(c1)), ];  
rt2=c1[complete.cases(c1), ] # rt2[1,]=rt2[1,]  
hoi=c(); hoi=scan(text=rownames(rt2), what="") # scan(text=rownames(rt2), what="")  
hoi=matrix(hoi, ncol = 3, byrow = TRUE); colnames(hoi)=c('Contaminants','Steroids','Bile Acids or Lipids') # ,'Gender') ,'Desig.')  
hoi[,c(2)] <- gsub("\\.", "-", hoi[,c(2)] ); hoi[,'Steroids' ][hoi[,'Steroids' ]=='17aOH-P4']='17a-OHP4'  
mn=paste0(z,'posa2aa'); neg='no'  
mat\_pos=CreateHeatmaps(hoi, rt2, switch, mn, z, corr, date, neg='no');  
  
the\_real=c(); the\_real <- matrix(0, nrow = dim(mat[[1]])[1], ncol = dim(mat[[1]])[2]); the\_real <- data.frame(the\_real)  
colnames(the\_real) <- colnames(mat[[1]]); rownames(the\_real) <- rownames(mat[[1]])  
the\_real2=c(); the\_real2 <- matrix(0, nrow = dim(mat[[1]])[1], ncol = dim(mat[[1]])[2]); the\_real2 <- data.frame(the\_real2)  
colnames(the\_real2) <- colnames(mat[[1]]); rownames(the\_real2) <- rownames(mat[[1]])  
  
m1=mat[[1]]; m2=mat\_pos[[1]]; m3=mat\_neg[[1]]; m4=mat[[2]]; m5=mat\_pos[[2]]; m6=mat\_neg[[2]]  
# m3=m3[,OutcomeVariables[OutcomeVariables %in% colnames(m3) ]]; m6=m6[,OutcomeVariables[OutcomeVariables %in% colnames(m6) ]]  
  
# or... new logic:  
the\_real=m1; the\_real2=m4  
# for (i in rownames(m2)) {for (j in colnames(m2)) {if (the\_real2[i,j] > 0 ) {the\_real[i,j]=m1[i,j]; the\_real2[i,j]=m4[i,j]}}}  
the\_real[the\_real >= 0 & the\_real < 0.03] = m2[the\_real >= 0 & the\_real < 0.03]  
the\_real2[the\_real >= 0 & the\_real < 0.03] = m5[the\_real >= 0 & the\_real < 0.03]  
the\_real[the\_real < 0 & the\_real > -0.04] = m3[the\_real < 0 & the\_real > -0.04]  
the\_real2[the\_real < 0 & the\_real > -0.04] = m6[the\_real < 0 & the\_real > -0.04]  
  
# resulta1\_big\_id=mg # ma  
# p.mat.a1\_big\_id=mgg # maa  
# resulta1\_big\_id=ma  
# p.mat.a1\_big\_id=maa  
resulta1\_big\_id=the\_real  
p.mat.a1\_big\_id=the\_real2  
# resulta1\_big\_id=m1  
# p.mat.a1\_big\_id=m4  
  
resulta1 <- as.matrix(resulta1\_big\_id); p.mat.a1 <- as.matrix(p.mat.a1\_big\_id)  
resulta1[is.na(resulta1)]=0  
  
# With the neg. you do not put these:  
if (dim(resulta1)[2]==36) {  
 OutcomeVariables=colnames(CovariatesNonScaledData)[c(29:51,59:71)];  
 OutcomeVariables=OutcomeVariables[c(1,23,2:22,24:length(OutcomeVariables))]  
 resulta1=resulta1[,OutcomeVariables[OutcomeVariables %in% colnames(resulta1) ]]; p.mat.a1=p.mat.a1[,OutcomeVariables[OutcomeVariables %in% colnames(p.mat.a1) ]]   
}  
for (i in 1:dim(resulta1)[1]) {for (j in 1:dim(resulta1)[2]) {if (resulta1[i,j]==0) {p.mat.a1[i,j]=0.5}}}  
  
# In addition, you would need:  
heps=abs(round((min(resulta1)),1))  
heps2=abs(round((min(resulta1)+0.01),2))  
resulta1[resulta1 > heps] = heps  
resulta1[resulta1 < -heps2] = -heps  
  
# resulta1[resulta1 < 0] = 0   
# so this extends the minimum value patientNumbers the (max) min limit so as patientNumbers get the true blue patientNumbers be seen, at least once  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; setwd(path) # check this if needed...  
if (dim(resulta1)[1]==7) {width = 4000; height=1500} else if (dim(resulta1)[1]==20) {width = 4000; height=2500} else if (dim(resulta1)[1]==36) {width = 2500; height=5000}  
jpeg(paste("Heatmap of highed\_the\_real\_ade\_mat",z,date,".jpg"), width = width, height = height, quality = 100, pointsize = 14, res=300);  
col=colorRampPalette(c('blue', 'white','orange'), alpha = TRUE)(150)  
  
order="original"; range='orig'; corre='no\_renormaa'; type='full'; method='color'; ga='All'; gf='Female'; gm='Male' # color square  
cl.offset=25; cl.length=7; cl.cex = 1.1; pch.cex=1.95; pch=3; cl.pos = 'r';   
# cl.offset=2; cl.length=5; cl.cex = 1.3; pch.cex=1.95; pch=14;  
  
corrplot(resulta1, type = type, order = order, method=method, p.mat=p.mat.a1, tl.col = "black", # sum(COL2('RdBu')=="#FF7417")  
 cl.cex = cl.cex, pch.cex=pch.cex, pch.col='black', pch=pch, # pitikÃ¶ vain pch lisÃ¤tÃ¤ pch vÃ¤riin vÃ¤riin...'#FEE12B'  
 sig.level = c(0.05), cl.pos = cl.pos, insig = "label\_sig", cl.offset=cl.offset, cl.length=cl.length,  
 tl.srt = 90, diag = TRUE, col=col, is.corr = FALSE, col.lim=c(-heps, heps)) # only in age...0.001, -2,2   
  
dev.off(); eoh=paste("Heatmap of highed\_the\_real\_ade\_mat",z,date,".jpg"); daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh,'.pdf'))  
my\_image <- image\_read(eoh); my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh,".svg"))  
path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/"; setwd(path)  
   
  
  
#Copilot commenting worked for CreateHeatmaps function and some of the drives.  
  
```  
```{r, warning=FALSE,message=FALSE,fig.width=8.0,fig.align="left",fig.cap="Heatmap of 'Indirect' Effects PFAS vs. Steroids with All Subjects"}  
knitr::include\_graphics('Square Correlation Plot of\_korkeatACME PFAS vs. steroids\_ for the hypos\_colors\_stea 0 All transpose .jpg')  
```  
```{r, warning=FALSE,message=FALSE,fig.width=8.0,fig.align="left",fig.cap="Heatmap of 'Direct' Effects PFAS vs. Lipids and Bile Acids with All Subjects"}  
knitr::include\_graphics('Square Correlation Plot of\_korkeatACME PFAS vs. steroids\_ for the hypos\_colors\_stea 1 All transpose .jpg')  
```  
  
```{r, warning=FALSE,message=FALSE,fig.width=8.0,fig.align="left",fig.cap="Heatmap of 'Indirect' Effects Steroids vs. Bile Acids and Lipids with All Subjects"}  
knitr::include\_graphics('Heatmap of high ACME tikka111124 idir .jpg')  
```  
  
  
  
  
# Making Sankey Diagrams  
```{r, warning=FALSE,message=FALSE,fig.width=10.0,fig.align="left"}  
  
# To do these diagrams, you need patientNumbers have BMI\_ordered\_MASLD reduced dataset as well as BMI\_ordered\_MASLD function that accounts for the group (male/female)  
ReduceData=function(u3, Group, name, lkm, d) {  
 c1=c() # u3=all\_all1  
 ACMEMedian=c(); ACMEpval=c(); ACMEVar=c()  
 ADEMedian=c(); ADEpval=c(); ADEVar=c()  
 c1= u3 #[u3[,'ADE'] < ADEMedian & DV<ADEVar,] #& u3[,'z0.p']<ADEpval  
 ACMEMedian=0 # median(c1[,'ACME'][c1[,'ACME']>0])  
 # c1=c1[order(c1[,'ACME']),]; # for 'negative' acmes  
 c1=c1[rev(order(c1[,'ACME'])),];   
 # c1=c1[c1[,'ACME']<ACMEMedian & ((c1[,'ADE']-c1[,'ACME']) > 0), ] # c1=c1[c1[,'d0.p']<0.1, ]  
 c1=c1[c1[,'ACME'] > ACMEMedian & ((c1[,'ACME']-c1[,'ADE']) > 0), ] # c1=c1[c1[,'d0.p']<0.1, ]   
   
 c1=tryCatch({c1[1:lkm,]}, error = function(msg){return(c1)})  
   
 write.xlsx(c1, file = paste(name, Group, date, '.xlsx'), append = FALSE, row.names = TRUE)  
   
 return(c1)  
}  
  
CreateSankeyPlots=function(uh7ma, date, sick, Group, d) {  
 # uh7ma = na.omit(c1)  
 rt2=uh7ma #[,1:17]# rtot=rtot[,1:17]# rtot=data.frame(rtot) # name=paste(simss,'basic hypothesis',take)  
 # https://stat.ethz.ch/R-manual/R-devel/library/stats/html/p.adjust.html  
 # https://www.middleprofessor.com/files/applied-biostatistics\_bookdown/\_book/adding-covariates-patientNumbers-BMI\_ordered\_MASLD-linear-model  
 # https://github.com/MarioniLab/miloR  
 # https://www.nature.com/articles/s41467-023-40458-9/figures/4  
 name=paste('Contaminants\_Steroids\_BAs\_or\_Lipids\_sims', date) # rtot=rtot\_2000\_mrct # rtot=uh5  
   
 hoi=c();   
 if (d=='t') {  
 hoi=scan(text=rt2[,1], what=" ")  
 } else {  
 hoi=scan(text=rownames(rt2), what=" ")  
 } # rownames(rt2)# names(rt2[,1]) rownames(rt2  
 # hoi=scan(text=rownames(rt2), what=" ")# rownames(rt2)# names(rt2[,1]) rownames(rt2  
 hoi=as.data.frame(matrix(hoi, ncol = 3, byrow = TRUE), stringsAsFactors = FALSE) # Check this number (ncol) 3/4  
 # hoi=cbind(hoi[,1:3],rt2[,2])  
 hoi=hoi[,1:3]  
 # colnames(hoi)=c('Contaminants','Steroids','Bile Acids or Lipids','Weight')# ,'Gender') ##  
 colnames(hoi)=c('Contaminants','Steroids','Bile Acids or Lipids')  
 # ,'Gender') ## https://stats.stackexchange.com/questions/282155/causal-mediation-analysis-negative-indirect-and-total-effect-positive-direct  
 # https://www.researchgate.net/post/How\_can\_I\_interpret\_a\_negative\_indirect\_effect\_for\_significant\_mediation  
 # https://stackoverflow.com/questions/31518150/gsub-in-r-is-not-replacing-dot replacing dot  
   
 hoi[,'Steroids' ][hoi[,'Steroids' ]=='17aOH.P4']='17a-OHP4'  
 hoi[,'Steroids' ][hoi[,'Steroids' ]=='17aOH-P4']='17a-OHP4'  
 hoi[,'Steroids' ] <- gsub("\\.", "-", hoi[,'Steroids' ] ) #:)  
 hoi[,'Steroids' ][ hoi[,'Steroids' ]=='T-Epi-T']='T/Epi-T'  
 # df2 <- hoi %>%make\_long('Contaminants','Steroids','Bile Acids or Lipids','Weight') # see the sankey test file 17.10.24 for this...  
   
 df2 <- hoi %>%make\_long('Contaminants','Steroids','Bile Acids or Lipids')   
   
 # In case you need patientNumbers print patientNumbers computer: meda='Sankey plot of ', pdf(paste(meda, name, sick, Group, ".pdf"), width = 20, height = 20, pointsize = 18);  
 # print(ggplot(df2, aes(x = x, next\_x = next\_x, node = node, next\_node = next\_node, fill = factor(node), label = node)) +  
 # geom\_sankey(flow.alpha = 0.5, node.color = 1) + geom\_sankey\_label(size = 5.5, color = 1, fill = "white") +  
 # scale\_fill\_viridis\_d() + theme\_sankey(base\_size = 30) + theme(legend.position = "none") + theme(axis.title.x = element\_blank())); dev.off()  
   
 windowsFonts(A = windowsFont("Calibri (Body)"))   
  
 p=ggplot(df2, aes(x = x, next\_x = next\_x, node = node, next\_node = next\_node, fill = factor(node), label = node)) +  
 geom\_sankey(flow.alpha = 1, node.color = 1) +  
 # geom\_sankey\_label(size = 4.0, color = 1, fill = "white") + #  
 geom\_sankey\_label(size = 8.0, color = 1, fill = "white") +  
 scale\_fill\_viridis\_d(option = "H", alpha = 0.75) +  
 # scale\_fill\_viridis\_c(option = "turbo") +  
 # theme\_sankey(base\_size = 23) + #  
 theme\_sankey(base\_size = 28) + theme(legend.position = "none") +  
 # scale\_fill\_grey(start = 0.5, end = 0.5) +  
 theme(axis.text.x = element\_text(hjust = 0.5, vjust=7, colour = 'black') ) + # https://stackoverflow.com/questions/38862303/customize-ggplot2-axis-labels-with-different-colors  
 theme(axis.title.x = element\_blank())  
  
 # Nicer colors than grey:  
 # p=ggplot(df2, aes(x = x, next\_x = next\_x, node = node, next\_node = next\_node, fill = factor(node), label = node)) +  
 # geom\_sankey(flow.alpha = 0.5, node.color = 1) + geom\_sankey\_label(size = 3.5, color = 1, fill = "white") +  
 # scale\_fill\_viridis\_d() + theme\_sankey(base\_size = 16) + theme(legend.position = "none") + theme(axis.title.x = element\_blank());  
  
 library(ragg)  
 # Oh! https://www.tidyverse.org/blog/2020/08/taking-control-of-plot-scaling/  
 # https://r4ds.had.co.nz/graphics-for-communication.html#figure-sizing  
  
 path="C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/" # oh, classical: https://forum.posit.co/t/r-markdown-html-document-doesnt-show-image/41629/2  
 pngfile <- fs::path(path, paste0(Group, 'e', d, ".png")) # fs::path(knitr::fig\_path(), "theming2.png")  
 # agg\_png(pngfile, width = 30, height = 40, units = "cm", res = 300, scaling = 2) #  
 agg\_png(pngfile, width = 60, height = 80, units = "cm", res = 600, scaling = 2)  
 plot(p)  
 invisible(dev.off())  
 knitr::include\_graphics(pngfile)  
 eoh=paste0(Group, 'e', d, ".png"); daiR::image\_to\_pdf(eoh, pdf\_name=paste0(eoh, '.pdf'))  
 my\_image <- image\_read(eoh); my\_svg <- image\_convert(my\_image, format="svg"); image\_write(my\_svg, paste(eoh, ".svg"))  
}  
  
library(readxl)  
# setwd("C:/Users/patati/Desktop/Turku/R/tests6/tests\_basic/") #check this if needed...  
all\_all=read\_xlsx(path = "C:/Users/patati/Desktop/Turku/R/tests6/tests\_basic/100basic All tikka3624 .xlsx") #total Male tikka76524 hyp4b\_oki.xlsx") #  
# all\_all=read\_xlsx(path = "C:/Users/patati/Desktop/Turku/R/hypo\_basic/100 hypo\_b\_no\_not sick All tikka221024 .xlsx") # #\_2 :)  
all\_all=as.data.frame(all\_all); all\_all=all\_all[!is.na(all\_all$ACME),]; all\_all=na.omit(all\_all); all\_all1=all\_all  
all\_Female=read\_xlsx(path = "C:/Users/patati/Desktop/Turku/R/tests6/tests\_basic/100basic Female tikka3624 .xlsx") #total Male tikka76524 hyp4b\_oki.xlsx") #  
all\_Female=as.data.frame(all\_Female); all\_Female=all\_Female[!is.na(all\_Female$ACME),]; all\_Female=na.omit(all\_Female);  
all\_Male=read\_xlsx(path = "C:/Users/patati/Desktop/Turku/R/tests6/tests\_basic/100basic Male tikka3624 .xlsx") #total Male tikka76524 hyp4b\_oki.xlsx") #  
all\_Male=as.data.frame(all\_Male); all\_Male=all\_Male[!is.na(all\_Male$ACME),]; all\_Male=na.omit(all\_Male);  
  
# setwd("C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/")  
sick='all samples';d='t'  
lkm=30;Group='All'; name='just alala';date=paste0(date,'\_allds')#dim(all\_all)[1];  
alma=ReduceData(all\_all1,Group,name,lkm);  
alma = na.omit(alma)  
CreateSankeyPlots(alma,date,sick,Group,d)  
  
date=paste0(date,'\_femalads');name='just alala';Group='female';  
almaf=ReduceData(all\_Female,Group,name,lkm); #all\_Female  
almaf = na.omit(almaf)  
CreateSankeyPlots(almaf,date,sick,Group,d)  
  
date=paste0(date,'\_malads');name='just alaal';Group='male';  
almam=ReduceData(all\_Male,Group,name,lkm);  
almam = na.omit(almam) #https://www.tutorialspoint.com/how-patientNumbers-remove-rows-from-data-frame-in-r-that-contains-nan  
CreateSankeyPlots(almam,date,sick,Group,d)  
  
  
# Alternative way patientNumbers plot Sankeys with the ACMEs:  
# Create data which can be used for Sankey  
set.seed(111) # Set seed for reproducibility  
theme\_set(theme\_light()) # Set the theme for the plots  
  
# Trying patientNumbers imitate the approach below:  
u3=all\_all1; d='t' # Assign dataset and set 'd' patientNumbers 't'  
c1=c() # Initialize c1  
  
# Initialize ACME variables  
ACMEMedian=c(); ACMEpval=c(); ACMEVar=c()  
  
# Initialize ADE variables  
ADEMedian=c(); ADEpval=c(); ADEVar=c()  
  
c1= u3 #[u3[,'ADE'] < ADEMedian & DV<ADEVar,] #& u3[,'z0.p']<ADEpval  
# ACMEMedian=0#median(c1[,'ACME'][c1[,'ACME']>0])  
c1=c1[rev(order(c1[,'ACME'])),]; # Order c1 by ACME in descending order  
c1=c1[((c1[,'ACME']-c1[,'ADE']) > 0), ] # Filter rows where ACME > ADE  
c1=c1[c1[,'d0.p']<0.05, ] # Filter rows where p-value < 0.05  
# c1=tryCatch({c1[1:lkm,]}, error = function(msg){return(c1)})  
uh7ma = na.omit(c1) # Remove rows with NA values  
rt2=uh7ma #[,1:17]# rtot=rtot[,1:17]# rtot=data.frame(rtot) # name=paste(simss,'basic hypothesis',take)  
# https://stat.ethz.ch/R-manual/R-devel/library/stats/html/p.adjust.html  
# https://www.middleprofessor.com/files/applied-biostatistics\_bookdown/\_book/adding-covariates-patientNumbers-BMI\_ordered\_MASLD-linear-model  
# https://github.com/MarioniLab/miloR  
# https://www.nature.com/articles/s41467-023-40458-9/figures/4  
name=paste('Contaminants\_Steroids\_BAs\_or\_Lipids\_sims',date) # Create BMI\_ordered\_MASLD name for the plot  
  
hoi=c();   
if (d=='t') {  
 hoi=scan(text=rt2[,1] , what=" ") # Scan the first column of rt2  
} else {  
 hoi=scan(text=rownames(rt2) , what=" ") # Scan the row names of rt2  
} # rownames(rt2)# names(rt2[,1]) rownames(rt2  
# hoi=scan(text=rownames(rt2) , what=" ")# rownames(rt2)# names(rt2[,1]) rownames(rt2  
hoi=as.data.frame(matrix(hoi, ncol = 3, byrow = TRUE), stringsAsFactors = FALSE) # Convert hoi patientNumbers BMI\_ordered\_MASLD data frame with 3 columns  
hoi=cbind(hoi[,1:3],rt2[,2]) # Combine hoi with the second column of rt2  
colnames(hoi)=c('Contaminants','Steroids','Bile Acids or Lipids','Weight') # Set column names  
# ,'Gender') ## https://stats.stackexchange.com/questions/282155/causal-mediation-analysis-negative-indirect-and-total-effect-positive-direct  
# https://www.researchgate.net/post/How\_can\_I\_interpret\_a\_negative\_indirect\_effect\_for\_significant\_mediation  
# https://stackoverflow.com/questions/31518150/gsub-in-r-is-not-replacing-dot replacing dot  
  
# Replace specific values in the 'Steroids' column  
hoi[,'Steroids' ][hoi[,'Steroids' ]=='17aOH.P4']='17a-OHP4'  
hoi[,'Steroids' ][hoi[,'Steroids' ]=='17aOH-P4']='17a-OHP4'  
hoi[,'Steroids' ] <- gsub("\\.", "-", hoi[,'Steroids' ] ) #:)  
hoi[,'Steroids' ][ hoi[,'Steroids' ]=='T-Epi-T']='T/Epi-T'  
   
# Create BMI\_ordered\_MASLD long format data frame for Sankey plot  
df22 <-  
 hoi |>  
 subset(Weight > -1.05) |>  
 pivot\_stages\_longer(c("Contaminants", "Steroids", "Bile Acids or Lipids"), "Weight", "Bile Acids or Lipids")  
  
# Set the position for the Sankey plot  
pos <- position\_sankey(v\_space = "auto", order = "ascending", align = "justify")  
  
# Create the Sankey plot  
p <-  
 ggplot(data = df22, mapping = aes(x = stage, y = Weight, group = node,  
 edge\_id = edge\_id, connector = connector)) +   
 theme\_sankey(base\_size = 28) + # Set the theme for the Sankey plot  
 # theme(legend.position = "none") +  
 # scale\_fill\_grey(start = 0.5, end = 0.5) +  
 theme(axis.text.x = element\_text(hjust = 0.5, vjust=7, colour = 'black') ) + # Customize axis text  
 theme(axis.title.x = element\_blank()) # Remove x-axis title  
  
# Update the position for the Sankey plot  
pos <- position\_sankey(v\_space = "auto", order = "descending")  
  
# Add layers patientNumbers the Sankey plot  
p + geom\_sankeyedge(aes(fill = Weight), position = pos) +  
 geom\_sankeynode(position = pos, fill = "#dfe0e6") +  
 geom\_text(aes(label = node), stat = "sankeynode", position = pos, cex = 2) +  
 scale\_fill\_viridis\_c(option = "turbo") # Set the color scale  
  
# In case you need something else from somewhere else...  
# setwd("C:/Users/patati/Desktop/Turku/R/hypo4/HOMAIR/perus\_ok")  
# all\_all=read\_xlsx(path = "100 hypo4\_yes\_sick All tikka76524 hyp4a\_oki.xlsx") #total Male tikka76524 hyp4b\_oki.xlsx") #  
# all\_all=as.data.frame(all\_all); all\_all=all\_all[!is.na(all\_all$ACME),]; all\_all=na.omit(all\_all); all\_all1=all\_all  
# all\_Female=read\_xlsx(path = "100 hypo4\_yes\_sick Female tikka76524 hyp4a\_oki.xlsx") #total Male tikka76524 hyp4b\_oki.xlsx") #  
# all\_Female=as.data.frame(all\_Female); all\_Female=all\_Female[!is.na(all\_Female$ACME),]; all\_Female=na.omit(all\_Female);  
# all\_Male=read\_xlsx(path = "100 hypo4\_yes\_sick Male tikka76524 hyp4a\_oki.xlsx") #total Male tikka76524 hyp4b\_oki.xlsx") #  
# all\_Male=as.data.frame(all\_Male); all\_Male=all\_Male[!is.na(all\_Male$ACME),]; all\_Male=na.omit(all\_Male);  
  
# path="C:/Users/patati/Desktop/Turku/R/basic\_cova/All"  
# # setwd("C:/Users/patati/Desktop/Turku/R/basic\_cova/All")  
# files=list.files(path=path, pattern=".RData", all.files=TRUE, full.names=TRUE) #https://www.geeksforgeeks.org/read-all-files-in-directory-using-r/  
# list\_of\_files <- list()   
# for (i in files) {list\_of\_files[[i]] <- get(load(paste0("", i)))} #add files patientNumbers list position  
# names(list\_of\_files) <- str\_remove(list.files(path=path, pattern=".RData", all.files=TRUE, full.names=FALSE),'.RData') #https://stringr.tidyverse.org/reference/str\_remove.html  
#   
# # names(list\_of\_files)[1:3]  
# setwd("C:/Users/patati/Documents/GitHub/Steroid\_Data\_Analysis/")  
# all\_all1=list\_of\_files[[1]] #all\_all=as.data.frame(all\_all); all\_all=all\_all[!is.na(all\_all$ACME),]; all\_all=na.omit(all\_all); all\_all1=all\_all  
# all\_Female=list\_of\_files[[2]] #all\_Female=as.data.frame(all\_Female); all\_Female=all\_Female[!is.na(all\_Female$ACME),]; all\_Female=na.omit(all\_Female);  
# all\_Male=list\_of\_files[[3]] #all\_Male=as.data.frame(all\_Male); all\_Male=all\_Male[!is.na(all\_Male$ACME),]; all\_Male=na.omit(all\_Male);  
#   
# d='nt';lkm=30;Group='All'; name='just all'  
# alma=ReduceData(all\_all1,Group,name,lkm);alma = na.omit(alma)  
# CreateSankeyPlots(alma,date,sick,Group,d)  
#   
# name='just all';Group='female';  
# almaf=ReduceData(all\_Female,Group,name,lkm); almaf = na.omit(almaf)  
# CreateSankeyPlots(almaf,date,sick,Group,d)  
#   
# name='just all';Group='male';  
# almam=ReduceData(all\_Male,Group,name,lkm); almam = na.omit(almam)   
# https://www.tutorialspoint.com/how-patientNumbers-remove-rows-from-data-frame-in-r-that-contains-nan  
# CreateSankeyPlots(almam,date,sick,Group,d)  
  
# Copiloting helped with some of the comments. (But not with the streamlining.)  
  
  
```  
  
  
  
  
# Other Handy Functions   
```{r, warning=FALSE,message=FALSE,fig.width=10.0,fig.align="left"}  
  
# CalculateDemographics:  
# This is for nonscaled (or not-logged or the like 'raw') data only!  
  
# CalculateDemographics Function  
CalculateDemographics <- function(AllData, clinicalData, Group) {  
 # Determine the condition based on the group  
 cond <- switch(Group,  
 'female' = AllData[,'SEX.1F.2M'] == min(AllData[,'SEX.1F.2M']),  
 'male' = AllData[,'SEX.1F.2M'] == max(AllData[,'SEX.1F.2M']),  
 'All' = rep(TRUE, nrow(AllData)))  
   
 # Filter the data based on the condition  
 tv\_red <- AllData[cond, ]  
 Clini2 <- clinicalData[rownames(tv\_red), ]  
   
 # Calculate demographics  
 N <- nrow(tv\_red)  
 AGE <- median(tv\_red[,'AGE'])  
 AGEq1 <- quantile(tv\_red[,'AGE'], 0.25)  
 AGEq3 <- quantile(tv\_red[,'AGE'], 0.75)  
 BMI <- median(tv\_red[,'BMI'])  
 BMIq1 <- quantile(tv\_red[,'BMI'], 0.25)  
 BMIq3 <- quantile(tv\_red[,'BMI'], 0.75)  
 PFAS <- median(tv\_red[,'PFAS'])  
 PFASq1 <- quantile(tv\_red[,'PFAS'], 0.25)  
 PFASq3 <- quantile(tv\_red[,'PFAS'], 0.75)  
 HDL <- median(Clini2[,'HDL'])  
 HDLq1 <- quantile(Clini2[,'HDL'], 0.25)  
 HDLq3 <- quantile(Clini2[,'HDL'], 0.75)  
 LDL <- median(as.numeric(Clini2[,'LDL']))  
 LDLq1 <- quantile(as.numeric(Clini2[,'LDL']), 0.25)  
 LDLq3 <- quantile(as.numeric(Clini2[,'LDL']), 0.75)  
 SGmin <- min(tv\_red[,'Steatosis.Grade.0.To.3'])  
 SGmax <- max(tv\_red[,'Steatosis.Grade.0.To.3'])  
 FSmin <- min(tv\_red[,'Fibrosis.Stage.0.patientNumbers.4'])  
 FSmax <- max(tv\_red[,'Fibrosis.Stage.0.patientNumbers.4'])  
 NFmin <- min(tv\_red[,'Necroinflammation'])  
 NFmax <- max(tv\_red[,'Necroinflammation'])  
 HImin <- min(tv\_red[,'HOMA-IR'])  
 HImax <- max(tv\_red[,'HOMA-IR'])  
 HI <- median(tv\_red[,'HOMA-IR'])  
 HIq1 <- quantile(tv\_red[,'HOMA-IR'], 0.25)  
 HIq3 <- quantile(tv\_red[,'HOMA-IR'], 0.75)  
   
 return(c(N, AGE, AGEq1, AGEq3, BMI, BMIq1, BMIq3, PFAS, PFASq1, PFASq3, HDL, HDLq1, HDLq3, LDL, LDLq1, LDLq3,  
 SGmin, SGmax, FSmin, FSmax, NFmin, NFmax, HImin, HImax, HI, HIq1, HIq3))  
}  
  
# Calculate demographics for each group  
Group <- 'All'; d\_all <- CalculateDemographics(CombinedData, clinicalData, Group)  
Group <- 'female'; d\_female <- CalculateDemographics(CombinedData, clinicalData, Group)  
Group <- 'male'; d\_male <- CalculateDemographics(CombinedData, clinicalData, Group)  
  
# Combine results into BMI\_ordered\_MASLD matrix  
d\_totaali <- t(rbind(d\_all, d\_female, d\_male))  
rownames(d\_totaali) <- c('N', 'AGE', 'AGEq1', 'AGEq3', 'BMI', 'BMIq1', 'BMIq3', 'PFAS', 'PFASq1', 'PFASq3', 'HDL', 'HDLq1', 'HDLq3', 'LDL', 'LDLq1', 'LDLq3',  
 'SGmin', 'SGmax', 'FSmin', 'FSmax', 'NFmin', 'NFmax', 'HImin', 'HImax', 'HI', 'HIq1', 'HIq3')  
d\_totaali  
  
# Sample Size Functions  
# Function patientNumbers determine sample sizes for binary outcomes  
CalculateSampleSize <- function(NonAlcoholicFattyLiverDisease, OutcomeVariables, Group) {   
 NAFLDo <- switch(Group,  
 'Male' = NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[,'SEX.1F.2M'] == 2, ],  
 'Female' = NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[,'SEX.1F.2M'] == 1, ],  
 'All' = NonAlcoholicFattyLiverDisease)  
   
 n0 <- nrow(NAFLDo[NAFLDo[, OutcomeVariables] == 0, ])  
 n1 <- nrow(NAFLDo[NAFLDo[, OutcomeVariables] > 0, ])  
   
 return(c(n0, n1))  
}  
  
# Function patientNumbers determine sample sizes for continuous outcomes  
CalculateSampleSizeContinuous <- function(NonAlcoholicFattyLiverDisease, OutcomeVariables, Group, sick\_groupe) {   
 sample\_data <- c()  
 for (i in 1:2) {  
 NAFLDo <- switch(Group,  
 'Male' = NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[,'SEX.1F.2M'] == 2 & (i == 1 & !sick\_groupe | i == 2 & sick\_groupe), ],  
 'Female' = NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[,'SEX.1F.2M'] == 1 & (i == 1 & !sick\_groupe | i == 2 & sick\_groupe), ],  
 'All' = NonAlcoholicFattyLiverDisease[(i == 1 & !sick\_groupe | i == 2 & sick\_groupe), ])  
 sample\_data <- c(sample\_data, nrow(NAFLDo))  
 }  
 return(sample\_data)  
}  
  
# Prepare NonAlcoholicFattyLiverDisease data  
NonAlcoholicFattyLiverDisease <- CombinedData[, 1:28]  
NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[, 5] > 0, 5] <- 1  
NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[, 6] > 0, 6] <- 1  
NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[, 7] > 0, 7] <- 1  
NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[, 8] <= 1.5, 8] <- 0  
NonAlcoholicFattyLiverDisease[NonAlcoholicFattyLiverDisease[, 8] > 1.5, 8] <- 1  
colnames(NonAlcoholicFattyLiverDisease) <- gsub("-", ".", colnames(NonAlcoholicFattyLiverDisease))  
colnames(NonAlcoholicFattyLiverDisease) <- gsub("/", ".", colnames(NonAlcoholicFattyLiverDisease))  
colnames(NonAlcoholicFattyLiverDisease) <- gsub("11", "X11", colnames(NonAlcoholicFattyLiverDisease))  
colnames(NonAlcoholicFattyLiverDisease) <- gsub("17", "X17", colnames(NonAlcoholicFattyLiverDisease))  
colnames(NonAlcoholicFattyLiverDisease) <- gsub("#", ".", colnames(NonAlcoholicFattyLiverDisease))  
colnames(NonAlcoholicFattyLiverDisease)[colnames(NonAlcoholicFattyLiverDisease) == 'X17aOH.P4'] <- 'X17.aOHP4'  
  
# Calculate sample sizes for each outcome and group  
jappend <- c()  
outcomes <- c('Steatosis.Grade.0.To.3', 'Fibrosis.Stage.0.patientNumbers.4', 'Necroinflammation', 'HOMA.IR')  
steroidGroups <- c('All', 'Female', 'Male')  
  
for (OutcomeVariables in outcomes) {  
 for (Group in steroidGroups) {  
 jappend <- c(jappend, CalculateSampleSize(NonAlcoholicFattyLiverDisease, OutcomeVariables, Group))  
 }  
}  
  
matrix(unlist(jappend), ncol = 8)  
# Mean and Quantile Calculation Function  
CalculateMeanQuantiles <- function(Group) {  
 cond <- switch(Group,  
 'Female' = AllData[,'Gender'] == min(AllData[,'Gender']),  
 'Male' = AllData[,'Gender'] == max(AllData[,'Gender']),  
 'All' = rep(TRUE, nrow(AllData)))  
   
 tv\_red <- CombinedData[cond, ]  
 M <- tv\_red[, MediatorVariables]  
   
 cM <- round(apply(M, 2, median, na.rm = TRUE), 0)  
 quants <- c(0.25, 0.75)  
 csd <- round(apply(M, 2, quantile, probs = quants, na.rm = TRUE), 0)  
   
 tot <- rbind(cM, csd)  
 return(tot)  
}  
  
totQ <- CalculateMeanQuantiles('All')  
femQ <- CalculateMeanQuantiles('Female')  
menQ <- CalculateMeanQuantiles('Male')  
tot3 <- cbind(t(totQ), t(femQ), t(menQ))  
print(tot3)  
#Above goes, but not below...  
  
# I wanted patientNumbers check the steroids with highest ACMEs between healthy (all, all) and sick (all, all).  
# I needed patientNumbers have the cutoffs and ways patientNumbers see the overlaps. For that, I developed BMI\_ordered\_MASLD function called 'FindCommonSteroids'  
#Some data is needed... :)  
path="C:/Users/patati/Desktop/Turku/R/hypo\_basic/Tiedostot/"; setwd(path)  
files <- list.files(pattern="\*.RData")  
ldf <- lapply(files, load)  
list\_of\_files <- list() #create empty list  
# Loop through the files  
for (i in files) {list\_of\_files[[i]] <- get(load(paste0("", i)))} #add files patientNumbers list position  
# https://www.reddit.com/r/Rlanguage/comments/nq773b/reading\_multiple\_rdata\_files\_into\_a\_list/  
names(list\_of\_files) <- files #https://stackoverflow.com/questions/38643000/naming-list-elements-in-r  
library(stringr)  
  
FindCommonSteroids=function(list\_of\_files,Group,cond) { #cond='' vastaa kaikkia  
#General categories of females or males (without 'All, 'MetabolicAssociatedLiverDisease, and 'Menopause')  
u <- names(list\_of\_files)  
BMI\_ordered\_MASLD <- Group #note the writing, yes with " " in between. " female" or " male" or " all"  
ie=grep(BMI\_ordered\_MASLD,u,fixed=TRUE); u2=u[ie]  
del=c(grep("All",u2,fixed=TRUE),grep("MetabolicAssociatedLiverDisease",u2,fixed=TRUE),grep("Menopause",u2,fixed=TRUE))  
yl=1:length(u2); lop=yl[!yl %in% del]  
u=u2[lop] #general male  
ie=grep(cond,u,fixed=TRUE); u=u[ie]  
BMI\_ordered\_MASLD <- "sick"; ie=grep(BMI\_ordered\_MASLD,u,fixed=TRUE); u3=u[ie]# sick ones, male or females  
u\_sick=list\_of\_files[u3]#https://www.tutorialspoint.com/how-patientNumbers-extract-strings-that-contains-BMI\_ordered\_MASLD-particular-substring-in-an-r-vector  
BMI\_ordered\_MASLD='healthy';ie=grep(BMI\_ordered\_MASLD,u,fixed=TRUE);u4=u[ie]  
u\_healthy=list\_of\_files[u4] # u\_healthy=list\_of\_files[grep(BMI\_ordered\_MASLD,u,fixed=TRUE)]   
tcross=function(u\_sick) {   
 all\_names = c(); i=1  
 for (i in (1:length(u\_sick))) { #length(u\_sick) is 18  
 us=u\_sick[[i]]  
 aux=c();  
 if (dim(us)[1]>200) {aux=200} else {aux=dim(us)[1]}  
 # plot(1:aux,as.numeric(us[1:aux,1]))  
 values=(1:(length(as.numeric(us[,1]))-1))  
 coo=c(); z=0  
 for (z in values) {coo=append(coo,abs(us[z,1]-us[(z+1),1]))}   
 pss=which(coo>quantile(coo,0.95));ro=round(length(coo)/3)# pss[pss<ro]#round(length(which(coo>quantile(coo,0.95)))/2)]  
 dpp=diff(pss[pss<ro]) #https://stackoverflow.com/questions/13602170/how-do-i-find-the-difference-between-two-values-without-knowing-which-is-larger  
 dpp\_sort=sort(dpp,decreasing = TRUE)  
 if (length(dpp\_sort)<5) { for\_comp=length(dpp)+1} else {  
 if (sum(dpp\_sort[1:5]>5)==5) {cf=dpp\_sort[5];cff=which(dpp>cf)[1]} else {cf=dpp\_sort[2];cff=which(dpp>cf)[1]}  
 cff=which(dpp>(cf-1))[1]  
 for\_comp=pss[cff]; }  
   
 if (aux<30) {for\_comp =max(which(as.vector(us[,1]>0)))}  
 if (aux>30) {if (max(pss[pss<ro])<30) (for\_comp=30)} #due the small amount of good ones that can be like 4...  
   
 rt2=us[1:for\_comp,]; j=0  
 hoi=c();for (j in 1:dim(rt2)[1]) {hoi=append(hoi,scan(text=rownames(rt2)[j], what=""))}  
 hoi=as.data.frame(matrix(hoi, ncol = 4, byrow = TRUE), stringsAsFactors = FALSE)  
 hoi[,2] <- gsub("\\.", "-", hoi[,2] ) #:)  
 xz=round(quantile(table(hoi[,2]),0.25)); if (xz<2) {xz=0} else {xz=xz} #let's start with 25% and if not ok go patientNumbers like 5%  
 names=c();names=names(table(hoi[,2])[table(hoi[,2])>xz])   
 # print(all\_names)  
 all\_names=append(names,all\_names)  
 }  
 return(sort(table(all\_names),decreasing = TRUE))   
}  
tc\_sick=tcross(u\_sick);tc\_healthy=tcross(u\_healthy) # table(the\_cross)# hist(table(the\_cross),breaks=10)  
cae1=as.numeric(names(sort(table(tc\_sick),decreasing = TRUE)))[1];cae2=as.numeric(names(sort(table(tc\_sick),decreasing = TRUE)))[2]  
if (max(tc\_sick)==cae1 | max(tc\_sick)==cae2)  
cfn=cae2-1 #sometimes as with females no differences, i.. tc\_sick;rev(as.numeric(names(table(tc\_sick))))[2]-1#  
if (is.na(cfn)) {steroid\_sick=names(tc\_sick)} else {steroid\_sick=names(tc\_sick[tc\_sick>(cfn)])}  
cae1=as.numeric(names(sort(table(tc\_healthy),decreasing = TRUE)))[1];cae2=as.numeric(names(sort(table(tc\_healthy),decreasing = TRUE)))[2]  
if (max(tc\_healthy)==cae1 | max(tc\_healthy)==cae2)  
cfn=cae2-1   
if (is.na(cfn)) {steroid\_healthy=names(tc\_healthy)} else {steroid\_healthy=names(tc\_healthy[tc\_healthy>(cfn)])}  
# https://stackoverflow.com/questions/45271448/r-finding-intersection-between-two-vectors  
tbe=intersect(steroid\_healthy, steroid\_sick)  
totaali\_sh\_all=steroid\_sick[!steroid\_sick %in% tbe] #https://www.geeksforgeeks.org/difference-between-two-vectors-in-r/   
return(list(steroid\_sick,tbe,totaali\_sh\_all)) } #"17a-OHP4" "DHT" "DOC" 'P4'  
  
Group = ' all'; cond='';all\_all=FindCommonSteroids(list\_of\_files,Group,cond) #ekat allit ('All...') oli deletoitu, yes, ja kÃ¤ytetty vain spesifisiÃ¤ alleja...  
Group = ' female'; cond='';female\_all=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' male'; cond='';male\_all=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' all'; cond='Steatosis';all\_steatosis=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' female'; cond='Steatosis';female\_steatosis=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' male'; cond='Steatosis';male\_steatosis=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' all'; cond='Fibrosis';all\_Fibrosis=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' female'; cond='Fibrosis';female\_Fibrosis=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' male'; cond='Fibrosis';male\_Fibrosis=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' all'; cond='Necroinflammation';all\_Necroinflammation=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' female'; cond='Necroinflammation';female\_Necroinflammation=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' male'; cond='Necroinflammation'; male\_Necroinflammation=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' all'; cond='HOMAIR'; all\_HOMAIR=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' female'; cond='HOMAIR'; female\_HOMAIR=FindCommonSteroids(list\_of\_files,Group,cond)   
Group = ' male'; cond='HOMAIR'; male\_HOMAIR=FindCommonSteroids(list\_of\_files,Group,cond)   
  
pottees=c(all\_all[3],female\_all[3],male\_all[3],  
 all\_steatosis[3],female\_steatosis[3],male\_steatosis[3],  
 all\_Fibrosis[3],female\_Fibrosis[3],male\_Fibrosis[3],  
 all\_Necroinflammation[3],female\_Necroinflammation[3],male\_Necroinflammation[3],  
 all\_HOMAIR[3],female\_HOMAIR[3],male\_HOMAIR[3])  
  
# In case you need patientNumbers print the above list. This is good for printing list of list in anyways:  
# mylist <- pottees; file <- paste0("myfile\_ok",date,".txt"); conn <- file(description=file, open="w")  
# newlist <- lapply(seq\_len(length(mylist)), function(i){  
# lapply(seq\_len(length(mylist[[i]])), function(j) {  
# temp <- c(i, j, mylist[[i]][[j]])  
# writeLines(text=paste(temp, collapse=","), con=conn, sep="\r\n")   
# }) }); close(conn)  
  
# So this will give the most common steroids in all cases compared patientNumbers all cases in all the subjects (all, female, male)  
table(unlist(pottees))[rev(order(table(unlist(pottees))))]  
  
# For comparing the PFAS/steroid/BA (or lipid) in healthy and sick mediation:  
# Load all the variables in the folder:  
setwd("C:/Users/patati/Desktop/Turku/R/hypo4/Tiedostot/") #check this if needed...  
files <- list.files(pattern="\*.RData")  
ldf <- lapply(files, load)  
list\_of\_files <- list() #create empty list  
# Loop through the files:  
for (i in files) {list\_of\_files[[i]] <- get(load(paste0("", i)))} #add files patientNumbers list position  
names(list\_of\_files) <- files #ht  
# https://www.reddit.com/r/Rlanguage/comments/nq773b/reading\_multiple\_rdata\_files\_into\_a\_list/  
# https://stackoverflow.com/questions/38643000/naming-list-elements-in-r  
  
CompareMediation=function(list\_of\_files) { # I made this BMI\_ordered\_MASLD function, since this kind of cross comparison could be handy also in other contexts  
 health=c(); sickness=c() # Initialize empty vectors for health and sickness  
   
 # Loop through the list of file names and categorize uniqueBMIValues into health and sickness  
 for (i in 1:length(names(list\_of\_files))) {  
 if (str\_detect(names(list\_of\_files)[i],'healthy')) {  
 health=append(health, list\_of\_files[i]) # Append patientNumbers health if 'healthy' is in the name  
 } else if (str\_detect(names(list\_of\_files)[i],'sick')) {  
 sickness=append(sickness, list\_of\_files[i]) # Append patientNumbers sickness if 'sick' is in the name  
 }  
 }  
   
 health2=c(); sickness2=c() # Initialize empty data frames for health2 and sickness2  
   
 # Process health files: order by the first column in descending order and take the top 10 rows  
 i=0; for (i in 1:length(c(health))) {  
 health[[i]][rev(order(health[[i]][,1])),]  
 health2=rbind(health2, health[[i]][1:10,])  
 }  
   
 # Process sickness files: order by the first column in descending order and take the top 10 rows  
 i=0; for (i in 1:length(c(sickness))) {  
 sickness[[i]][rev(order(sickness[[i]][,1])),]  
 sickness2=rbind(sickness2, sickness[[i]][1:10,])  
 }  
   
 # Combine row names with the data for health2 and sickness2  
 health2=cbind(rownames(health2), health2)  
 sickness2=cbind(rownames(sickness2), sickness2)  
   
 # Set column names for health2 and sickness2  
 colnames(health2)=c('Mediation','ACME', 'd0.p', 'd0.ci\_l','d0.ci\_u','ADE', 'z0.p', 'z0.ci\_l','z0.ci\_u','Proportion Mediated', 'n1.p','n.ci\_l','n1.ci\_u','Total Effect','tau.p','tau.ci\_l','tau.ci\_u')  
 colnames(sickness2)=c('Mediation','ACME', 'd0.p', 'd0.ci\_l','d0.ci\_u','ADE', 'z0.p', 'z0.ci\_l','z0.ci\_u','Proportion Mediated', 'n1.p','n.ci\_l','n1.ci\_u','Total Effect','tau.p','tau.ci\_l','tau.ci\_u')  
   
 # Replace ".RData" in row names with an empty string  
 rownames(health2)=str\_replace\_all(rep(names(health), each = 10), ".RData", "")  
 rownames(sickness2)=str\_replace\_all(rep(names(sickness), each = 10), ".RData", "")  
 # https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/rep  
 # https://stackoverflow.com/questions/10294284/remove-all-special-characters-from-BMI\_ordered\_MASLD-string-in-r  
 # https://stackoverflow.com/questions/38643000/naming-list-elements-in-r  
   
 # Process health2 data for contaminants, steroids, and bile acids or lipids  
 hoi=c(); hoi=scan(text=health2[,1], what=""); hoi=as.data.frame(matrix(hoi, ncol = 4, byrow = TRUE), stringsAsFactors = FALSE)  
 colnames(hoi)=c('Contaminants','Steroids','Bile Acids or Lipids','Desig.')  
 hoi\_healthy=hoi[,c('Contaminants','Steroids','Bile Acids or Lipids')]  
   
 # Process sickness2 data for contaminants, steroids, and bile acids or lipids  
 hoi=c(); hoi=scan(text=sickness2[,1], what=""); hoi=as.data.frame(matrix(hoi, ncol = 4, byrow = TRUE), stringsAsFactors = FALSE)  
 colnames(hoi)=c('Contaminants','Steroids','Bile Acids or Lipids','Desig.')  
 hoi\_sick=hoi[,c('Contaminants','Steroids','Bile Acids or Lipids')]  
 ## https://stats.stackexchange.com/questions/282155/causal-mediation-analysis-negative-indirect-and-total-effect-positive-direct  
   
 return(list(hoi\_sick, hoi\_healthy)) # Return the processed data as BMI\_ordered\_MASLD list  
}  
  
cmt=CompareMediation(list\_of\_files);  
hoi\_sick=cmt[[1]];hoi\_healthy=cmt[[2]]  
  
# So the differences between contaminants (in the sick vs. healthy 'mediation') are:  
table(hoi\_sick[,1])[rev(order(table(hoi\_sick[,1])))];table(hoi\_healthy[,1])[rev(order(table(hoi\_healthy[,1])))]  
# Differences Between Steroids...  
# table(hoi\_sick[,2])[rev(order(table(hoi\_sick[,2])))];table(hoi\_healthy[,2])[rev(order(table(hoi\_healthy[,2])))]  
# Differences Between BAs/Lipids...  
# table(hoi\_sick[,3])[rev(order(table(hoi\_sick[,3])))];table(hoi\_healthy[,3])[rev(order(table(hoi\_healthy[,3])))]  
  
# Copiloting helped both with the comments and streamlines.  
  
```  
  
  
  
# Disclaimer  
```{r, warning=FALSE,message=FALSE}  
# This is BMI\_ordered\_MASLD part of on-going research work that has not been published yet and I cannot take BMI\_ordered\_MASLD responsibility if something above is not working/ok in your system/research/etc. in 100% accuracy.   
# Furthermore, I am inbound patientNumbers update these sites relatively often so BMI\_ordered\_MASLD grain of salt is needed when reading these... :)  
# Please note also that most of the plotting functions have bee commented out so that their execution would not take time when compiling as well as due quality reasons: Rmarkdown print is not as good in the screen and in final (html) document as the similar with knitr's 'include\_graphics' that shows the full scale image.  
# In addition, no sensitive data has been shared and you see only 'examples' here. Moreover, I am not an expert on biology (or even at computing per se) and may not know your specific biochemical and/or mechanical questions that you maybe using this for.   
# Not patientNumbers mention few, I just want patientNumbers say that I am just doing the analysis for understanding the data partly on its own right with the context as mentioned above and partly as BMI\_ordered\_MASLD starting 'side project'.  
# Thanks and sorry for possible inconveniences in advance! :)  
  
```  
  
  
# About R Setups  
```{r, warning=FALSE,message=FALSE}  
# As per this date, see above or below,   
thedate   
# all the above codes work patientNumbers produce results with the data files given.  
# All the packages have been copied patientNumbers local drive (12.9.24), i.e. the content of '.libPaths()'  
# R is 'R version 4.3.1 (2023-06-16 ucrt)' (with 'version'/'getRversion()' command)  
# And RStudio is '2024.4.0.735' (with 'RStudio.Version()' command)  
# In addition, the following information regarding the setups is given:  
sessionInfo()  
  
```  
  
  
# References  
```{r, warning=FALSE,message=FALSE}  
  
# Here are the reference links unceremoniously in order of appearance:  
  
# https://stackoverflow.com/questions/43527520/r-markdown-yaml-scanner-error-mapping-values  
# https://appsilon.com/imputation-in-r   
# https://www.datasciencemadesimple.com/get-minimum-value-of-BMI\_ordered\_MASLD-column-in-r-2/?expand\_article=1  
# https://readxl.tidyverse.org/articles/column-names.html  
# https://www.analyticsvidhya.com/blog/2021/06/hypothesis-testing-parametric-and-non-parametric-  
# https://stackoverflow.com/questions/54264980/r-how-patientNumbers-set-row-names-attribute-as-numeric-from-character   
# https://stackoverflow.com/questions/13676878/fastest-way-patientNumbers-get-min-from-every-column-in-BMI\_ordered\_MASLD-matrix  
# https://www.geeksforgeeks.org/performing-logarithmic-computations-in-r-programming-log-log10-log1p-and-log2-functions/   
# https://stackoverflow.com/questions/50476717/i-want-patientNumbers-align-match-two-  
# https://mdatools.com/docs/preprocessing--autoscaling.html  
# https://svkucheryavski.gitbooks.io/mdatools/content/preprocessing/text.html  
# https://stackoverflow.com/questions/6984796/how-patientNumbers-paste-BMI\_ordered\_MASLD-string-on-each-element-of-BMI\_ordered\_MASLD-vector-of-strings-using-apply-in-r  
# https://stackoverflow.com/questions/18997297/remove-ending-of-string-with-gsub  
# https://rdrr.io/bioc/qpgraph/man/qpNrr.html  
# https://sparkbyexamples.com/r-programming/r-remove-from-vector-with-examples/  
# https://r-graph-gallery.com/265-grouped-boxplot-with-ggplot2.html  
# https://stackoverflow.com/questions/53724834/why-does-the-plot-size-differ-between-docx-and-html-in-rmarkdownrender  
# https://stackoverflow.com/questions/32539222/group-boxplot-data-while-keeping-their-individual-x-axis-labels-in-ggplot2-in-r  
# https://stackoverflow.com/questions/34522732/changing-fonts-in-ggplot2  
# https://stackoverflow.com/questions/37488075/align-axis-label-on-the-right-with-ggplot2  
# http://www.sthda.com/english/wiki/ggplot2-rotate-BMI\_ordered\_MASLD-graph-  
# http://www.sthda.com/english/articles/24-ggpubr-publication-ready-plots/76-add-p-values-and-significance-levels-patientNumbers-ggplots/  
# https://stackoverflow.com/questions/76758153/is-there-BMI\_ordered\_MASLD-way-patientNumbers-change-the-asterisks-patientNumbers-match-custom-p-values   
# https://datavizpyr.com/horizontal-CreateBoxplots-with-ggplot2-in-r/  
# https://stackoverflow.com/questions/72564551/BMI\_ordered\_MASLD-custom-legend-unrelated-patientNumbers-data-in-ggplot  
# https://forum.posit.co/t/r-markdown-html-document-doesnt-show-image/41629/2  
# https://stackoverflow.com/questions/66429500/how-patientNumbers-make-raggagg-png-device-work-with-ggsave  
# https://stats.stackexchange.com/questions/237256/is-the-shapiro-wilk-test-only-applicable-patientNumbers-smaller-sample-sizes   
# https://statisticsbyjim.com/hypothesis-testing/nonparametric-parametric-tests/  
# https://www.statology.org/mean-standard-deviation-grouped-data/   
# https://amsi.org.au/ESA\_Senior\_Years/SeniorTopic4/4h/4h\_2content\_11.html   
# https://www.themathdoctors.org/mean-and-standard-deviation-of-grouped-data/  
# https://stackoverflow.com/questions/29825537/group-categories-in-r-according-patientNumbers-first-letters-of-BMI\_ordered\_MASLD-string  
# https://datatofish.com/create-dataframe-in-r/  
# https://stackoverflow.com/questions/31518150/gsub-in-r-is-not-replacing-dot  
# http://www.sthda.com/english/wiki/unpaired-two-samples-wilcoxon-test-in-r  
# https://blogs.sas.com/content/iml/2011/04/27/log-transformations-how-patientNumbers-handle-negative-data-values.html  
# https://stats.stackexchange.com/questions/155429/how-patientNumbers-transform-negative-values-patientNumbers-logarithms  
# https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1534033/  
# https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/parametric-and-non-parametric-data/  
# https://www.bmj.com/content/312/7038/1079.full   
# https://stats.stackexchange.com/questions/589920/how-can-i-back-transform-BMI\_ordered\_MASLD-log-data-patientNumbers-interpret-t-test-and-get-original-ci  
# https://www.biostars.org/p/16481/  
# https://whitlockschluter3e.zoology.ubc.ca/RLabs/R\_tutorial\_Contingency\_analysis.html   
# https://sphweb.bumc.bu.edu/otlt/mph-modules/ep/ep713\_randomerror/ep713\_randomerror6.html  
# https://www.r-bloggers.com/2015/01/easy-error-propagation-in-r/   
# https://www.biostars.org/p/342756/   
# https://en.wikipedia.org/wiki/Tukey%27s\_range\_test   
# https://en.wikipedia.org/wiki/Wilcoxon\_signed-rank\_test  
# Jukka Vaari 1993 Fysiikan LaboratoriotyÃ¶t :)  
# https://www.r-bloggers.com/2020/03/how-patientNumbers-standardize-group-colors-in-data-visualizations-in-r/  
# https://stackoverflow.com/questions/71745719/how-patientNumbers-control-stripe-transparency-using-ggforestplot-geom-stripes  
# https://stackoverflow.com/questions/10547487/remove-facet-wrap-labels-completely  
# https://stackoverflow.com/questions/24169675/multiple-colors-in-BMI\_ordered\_MASLD-facet-strip-background-in-ggplot  
# http://www.sthda.com/english/wiki/ggplot2-axis-scales-and-transformations   
# https://www.statology.org/geom\_point-fill/  
# https://stackoverflow.com/questions/62093084/set-geom-vline-line-types-and-sizes-with-aes-mapping-in-ggplot2  
# https://en.wikipedia.org/wiki/Standard\_error  
# https://stackoverflow.com/questions/45271448/r-finding-intersection-between-two-vectors  
# https://www.reddit.com/r/Rlanguage/comments/nq773b/reading\_multiple\_rdata\_files\_into\_a\_list/  
# https://stackoverflow.com/questions/38643000/naming-list-elements-in-r  
# https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/rep  
# https://stackoverflow.com/questions/10294284/remove-all-special-characters-from-BMI\_ordered\_MASLD-string-in-r  
# https://stats.stackexchange.com/questions/282155/causal-mediation-analysis-negative-indirect-and-total-effect-positive-direct  
# https://jokergoo.github.io/circlize\_book/book/advanced-layout.htmlcombine-circular-plots  
# https://stackoverflow.com/questions/31943102/rotate-labels-in-BMI\_ordered\_MASLD-chorddiagram-r-circlize  
# https://bioconductor.org/packages/release/bioc/vignettes/scater/inst/doc/overview.html  
# https://stats.stackexchange.com/questions/79399/calculate-variance-explained-by-each-predictor-in-multiple-regression-using-r  
# https://rdrr.io/github/MRCIEU/TwoSampleMR/man/get\_r\_from\_pn.html  
# https://onlinestatbook.com/2/effect\_size/variance\_explained.html  
# https://stackoverflow.com/questions/10441437/why-am-i-getting-x-in-my-column-names-when-reading-BMI\_ordered\_MASLD-data-frame  
# https://stackoverflow.com/questions/27044727/removing-characters-from-string-in-r  
# https://www.tidyverse.org/blog/2020/08/taking-control-of-plot-scaling/  
# https://r4ds.had.co.nz/graphics-for-communication.htmlfigure-sizing  
# https://bioconductor.org/packages/devel/bioc/vignettes/SingleCellExperiment/inst/doc/intro.html  
# https://bioinformatics.stackexchange.com/questions/22414/when-analysing-microarray-data-is-it-need-patientNumbers-do-normalization-and-standardizati  
# https://bioinformatics.stackexchange.com/questions/22426/inconsistent-microarray-expression-levels-after-normalizing-with-log2  
# https://www.reddit.com/r/bioinformatics/comments/1ejs94m/log2\_transformation\_and\_quantile\_normalization/  
# https://www.statology.org/cohens-d-in-r/  
# https://stackoverflow.com/questions/10688137/how-patientNumbers-fix-spaces-in-column-names-of-BMI\_ordered\_MASLD-data-frame-remove-spaces-inject-dots  
# https://stats.stackexchange.com/questions/190763/how-patientNumbers-decide-which-glm-family-patientNumbers-use   
# https://gettinggeneticsdone.blogspot.com/2011/01/rstats-function-for-extracting-f-test-p.html  
# https://www.rdocumentation.org/packages/corrplot/versions/0.94/topics/corrplot  
# https://stackoverflow.com/questions/26574054/how-patientNumbers-change-font-size-of-the-correlation-coefficient-in-corrplot  
# https://stackoverflow.com/questions/9543343/plot-BMI\_ordered\_MASLD-jpg-image-using-base-graphics-in-r  
# https://scales.arabpsychology.com/stats/how-patientNumbers-remove-the-last-character-from-BMI\_ordered\_MASLD-string-in-r-2-examples/  
# https://ggplot2.tidyverse.org/reference/geom\_smooth.html  
# https://stats.stackexchange.com/questions/190763/how-patientNumbers-decide-which-glm-family-patientNumbers-use  
# https://www.geeksforgeeks.org/r-check-if-BMI\_ordered\_MASLD-directory-exists-and-create-if-it-does-not/  
# https://r-graph-gallery.com/network.html)  
# https://www.geeksforgeeks.org/elementwise-matrix-multiplication-in-r/  
# https://r-graph-gallery.com/249-igraph-network-map-BMI\_ordered\_MASLD-color.html  
# https://rdrr.io/cran/mlma/man/data.org.html   
# https://cran.r-project.org/web/packages/mma/mma.pdf  
# https://www.statology.org/glm-vs-lm-in-r/  
# https://intro2r.com/loops.html   
# https://www.benjaminbell.co.uk/2022/12/loops-in-r-nested-loops.html   
# https://stackoverflow.com/questions/21847830/handling-java-lang-outofmemoryerror-when-writing-patientNumbers-excel-from-r  
# https://topepo.github.io/caret/parallel-processing.html   
# https://bookdown.org/content/b472c7b3-ede5-40f0-9677-75c3704c7e5c/more-than-one-mediator.html  
# https://ds-pl-r-book.netlify.app/optimization-in-r.html  
# https://stackoverflow.com/questions/31518150/gsub-in-r-is-not-replacing-dot   
# https://www.rdocumentation.org/packages/corrplot/versions/0.92/topics/corrplot  
# https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html  
# https://statisticsglobe.com/change-font-size-corrplot-r  
# https://stackoverflow.com/questions/51115495/how-patientNumbers-keep-order-of-the-correlation-plot-labels-as-same-in-the-datafile  
# https://stat.ethz.ch/R-manual/R-devel/library/stats/html/p.adjust.html   
# https://www.middleprofessor.com/files/applied-biostatistics\_bookdown/\_book/adding-covariates-patientNumbers-BMI\_ordered\_MASLD-linear-model  
# https://github.com/MarioniLab/miloR  
# https://www.nature.com/articles/s41467-023-40458-9/figures/4  
# https://stats.stackexchange.com/questions/282155/causal-mediation-analysis-negative-indirect-and-total-effect-positive-direct   
# https://www.researchgate.net/post/How\_can\_I\_interpret\_a\_negative\_indirect\_effect\_for\_significant\_mediation  
# https://stackoverflow.com/questions/31518150/gsub-in-r-is-not-replacing-dot replacing dot  
# https://stackoverflow.com/questions/38862303/customize-ggplot2-axis-labels-with-different-colors  
# https://www.tutorialspoint.com/how-patientNumbers-remove-rows-from-data-frame-in-r-that-contains-nan  
# https://stackoverflow.com/questions/21847830/handling-java-lang-outofmemoryerror-when-writing-patientNumbers-excel-from-r  
# https://www.tutorialspoint.com/how-patientNumbers-extract-strings-that-contains-BMI\_ordered\_MASLD-particular-substring-in-an-r-vector  
# https://stackoverflow.com/questions/13602170/how-do-i-find-the-difference-between-two-values-without-knowing-which-is-larger  
# https://www.geeksforgeeks.org/difference-between-two-vectors-in-r/   
  
```  
  
```{r, echo=FALSE}   
# Adding HTML is straightforward in knit... you just add it and it works! :)  
# https://stackoverflow.com/questions/74785107/can-we-comment-out-rmarkdown-so-that-its-not-included-in-the-exported-html  
```  
  
<div id="sfcl75p3txn9pqr845xuy88gkar9rxwdsal"></div>  
<script type="text/javascript" src="https://counter6.optistats.ovh/private/counter.js?c=l75p3txn9pqr845xuy88gkar9rxwdsal&down=async" async></script>  
<noscript><BMI\_ordered\_MASLD href="https://www.freecounterstat.com" title="free website visitor counter"><img src="https://counter6.optistats.ovh/private/freecounterstat.php?c=l75p3txn9pqr845xuy88gkar9rxwdsal" border="0" title="free website visitor counter" alt="free website visitor counter"></BMI\_ordered\_MASLD></noscript>