IGT explore modeling

Eliany Perez

3/17/2021

setwd ("C:/Users/elianyperez/Desktop/Rmodeling")  
  
  
#to change settings for r to run go to https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started  
library(devtools)

## Loading required package: usethis

## Error in get(genname, envir = envir) : object 'testthat\_print' not found

library(hBayesDM)

## Loading required package: Rcpp

##   
##   
## This is hBayesDM version 1.0.2

library(rstan)

## Loading required package: StanHeaders

## Loading required package: ggplot2

## rstan (Version 2.21.2, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling  
## options(mc.cores = parallel::detectCores()).  
## To avoid recompilation of unchanged Stan programs, we recommend calling  
## rstan\_options(auto\_write = TRUE)

## Do not specify '-march=native' in 'LOCAL\_CPPFLAGS' or a Makevars file

library(ggplot2)  
library(ggpubr)  
library(ggcorrplot)

Run PVL model

#pvl<-igt\_pvl\_delta(data = "C:/Users/elianyperez/Desktop/Rmodeling/igt\_data2.txt", niter = 4000, nwarmup = 1000,  
 # nchain = 4, ncore = 1, nthin = 1, inits = "vb",  
 # indPars = "mean", modelRegressor = FALSE, vb = FALSE,  
 # inc\_postpred = FALSE, adapt\_delta = 0.95, stepsize = 1,  
 # max\_treedepth = 10)

# Visually check convergence of the sampling chains (should look like 'hairy caterpillars')  
#plot(pvl, type = "trace")  
  
# Check Rhat values (all Rhat values should be less than or equal to 1.1)  
#rhat(pvl)  
  
# Plot the posterior distributions of the hyper-parameters (distributions should be unimodal)  
#plot(pvl)  
  
# Show the WAIC and LOOIC model fit estimates  
#printFit(pvl)

Run orl model

#orl<-igt\_orl(data = "C:/Users/elianyperez/Desktop/Rmodeling/igt\_data2.txt", niter = 4000, nwarmup = 1000, nchain = 4,  
 # ncore = 1, nthin = 1, inits = "vb", indPars = "mean",  
 #modelRegressor = FALSE, vb = FALSE, inc\_postpred = FALSE,  
 #adapt\_delta = 0.95, stepsize = 1, max\_treedepth = 10)

# Visually check convergence of the sampling chains (should look like 'hairy caterpillars')  
#plot(orl, type = "trace")  
  
# Check Rhat values (all Rhat values should be less than or equal to 1.1)  
#rhat(orl)  
  
# Plot the posterior distributions of the hyper-parameters (distributions should be unimodal)  
#plot(orl)  
  
# Show the WAIC and LOOIC model fit estimates  
#printFit(orl)

Run vpp model

#vpp<-igt\_vpp(data = "C:/Users/elianyperez/Desktop/Rmodeling/igt\_data2.txt", niter = 4000, nwarmup = 1000, nchain = 4,  
 # ncore = 4, nthin = 1, inits = "vb", indPars = "mean",  
 #modelRegressor = FALSE, vb = FALSE, inc\_postpred = FALSE,  
 #adapt\_delta = 0.95, stepsize = 1, max\_treedepth = 10)

# Visually check convergence of the sampling chains (should look like 'hairy caterpillars')  
#plot(vpp, type = "trace")  
  
# Check Rhat values (all Rhat values should be less than or equal to 1.1)  
#rhat(vpp)  
  
# Plot the posterior distributions of the hyper-parameters (distributions should be unimodal)  
#plot(vpp)  
  
# Show the WAIC and LOOIC model fit estimates  
#printFit(vpp)  
  
#save.image(vpp.RData)

Correlations between mode parameters and raw E choice

setwd("G:/My Drive/EP\_Year1/IGT analysis/Rdata")  
load("models.RData")

## Registered S3 methods overwritten by 'lme4':  
## method from  
## cooks.distance.influence.merMod car   
## influence.merMod car   
## dfbeta.influence.merMod car   
## dfbetas.influence.merMod car

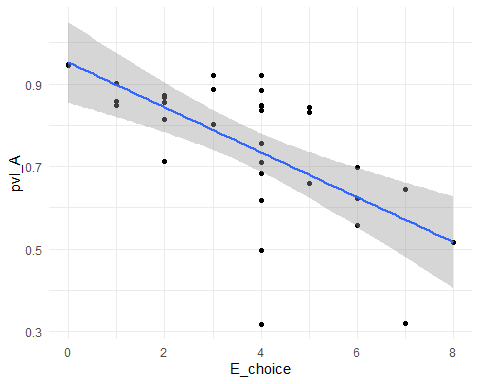
setwd("C:/Users/elianyperez/Desktop/Rmodeling")  
 pvlPars<-as.data.frame(pvl$allIndPars)  
 orlPars<-as.data.frame(orl$allIndPars)  
 vppPars<-as.data.frame(vpp$allIndPars)  
 Echoice<- read.csv("C:/Users/elianyperez/Desktop/Rmodeling/Etotal.csv")  
   
 names(pvlPars)<-c("ID","pvl\_A","pvl\_alpha","pvl\_cons","pvl\_lambda")  
 names(orlPars)<-c("ID","orl\_Arew","orl\_Apun","orl\_K","orl\_betaF","orl\_betaP" )  
 names(vppPars)<-c("ID","vpp\_A","vpp\_alpha","vpp\_cons","vpp\_lambda","vpp\_epP","vpp\_epN","vpp\_K","vpp\_w")  
   
 #merge- can only merge 2 at a time  
 merged<-merge(pvlPars,orlPars,by.x="ID")  
 merged2<-merge(vppPars,Echoice,by.x="ID")  
 mergedPars<-merge(merged,merged2,by.x="ID")  
  
names(mergedPars)

## [1] "ID" "pvl\_A" "pvl\_alpha" "pvl\_cons" "pvl\_lambda"  
## [6] "orl\_Arew" "orl\_Apun" "orl\_K" "orl\_betaF" "orl\_betaP"   
## [11] "vpp\_A" "vpp\_alpha" "vpp\_cons" "vpp\_lambda" "vpp\_epP"   
## [16] "vpp\_epN" "vpp\_K" "vpp\_w" "E\_choice" "Total"

#choose columns  
 mergedPars<- mergedPars[c("pvl\_A","pvl\_alpha","pvl\_cons","pvl\_lambda","orl\_Arew","orl\_Apun","orl\_K","orl\_betaF","orl\_betaP","vpp\_A","vpp\_alpha","vpp\_cons","vpp\_lambda","vpp\_epP","vpp\_epN","vpp\_K","vpp\_w","E\_choice")]   
   
 pvlPars<-mergedPars[c("pvl\_A","pvl\_alpha","pvl\_cons","pvl\_lambda","E\_choice")]   
 orlPars<-mergedPars[c("orl\_Arew","orl\_Apun","orl\_K","orl\_betaF","orl\_betaP","E\_choice")]  
 vppPars<-mergedPars[c("vpp\_A","vpp\_alpha","vpp\_cons","vpp\_lambda","vpp\_epP","vpp\_epN","vpp\_K","vpp\_w","E\_choice")]   
   
 #if you want to write the file  
 write.csv(mergedPars, "C:/Users/elianyperez/Desktop/Rmodeling/AllIndpars.csv", row.names=FALSE)

theme\_set(  
 theme\_minimal() +  
 theme(legend.position = "right")  
 )  
 a <- ggplot(mergedPars, aes(x = E\_choice, y = pvl\_A))  
 # Scatter plot with regression line  
 a + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'

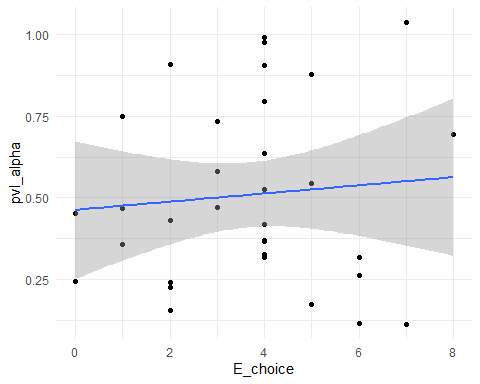


cor.test(mergedPars$E\_choice, mergedPars$pvl\_A,method = c("pearson", "kendall", "spearman"))

##   
## Pearson's product-moment correlation  
##   
## data: mergedPars$E\_choice and mergedPars$pvl\_A  
## t = -4.74, df = 31, p-value = 4.519e-05  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.8110519 -0.3922173  
## sample estimates:  
## cor   
## -0.6482397

b <- ggplot(mergedPars, aes(x = E\_choice, y = pvl\_alpha))  
 # Scatter plot with regression line  
 b + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'

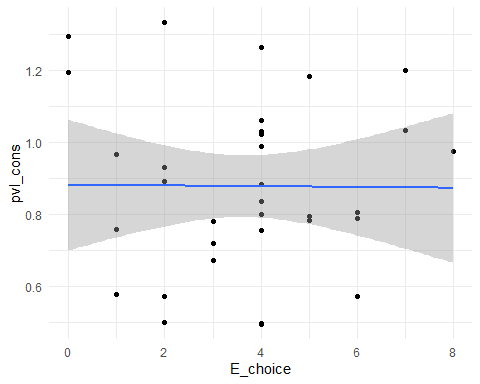


cor.test(mergedPars$E\_choice, mergedPars$pvl\_alpha,method = c("pearson", "kendall", "spearman"))

##   
## Pearson's product-moment correlation  
##   
## data: mergedPars$E\_choice and mergedPars$pvl\_alpha  
## t = 0.50427, df = 31, p-value = 0.6176  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.2611971 0.4204882  
## sample estimates:  
## cor   
## 0.09020032

c <- ggplot(mergedPars, aes(x = E\_choice, y = pvl\_cons))  
 # Scatter plot with regression line  
 c + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'

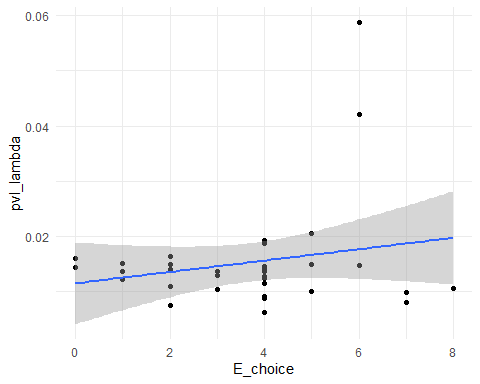


cor.test(mergedPars$E\_choice, mergedPars$pvl\_cons,method = c("pearson", "kendall", "spearman"))

##   
## Pearson's product-moment correlation  
##   
## data: mergedPars$E\_choice and mergedPars$pvl\_cons  
## t = -0.0494, df = 31, p-value = 0.9609  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.3511118 0.3354586  
## sample estimates:  
## cor   
## -0.008872214

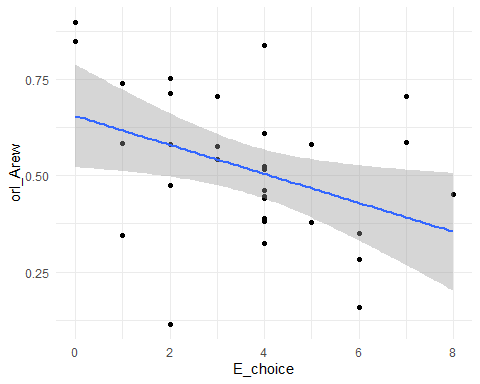
d <- ggplot(mergedPars, aes(x = E\_choice, y = pvl\_lambda))  
 # Scatter plot with regression line  
 d + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



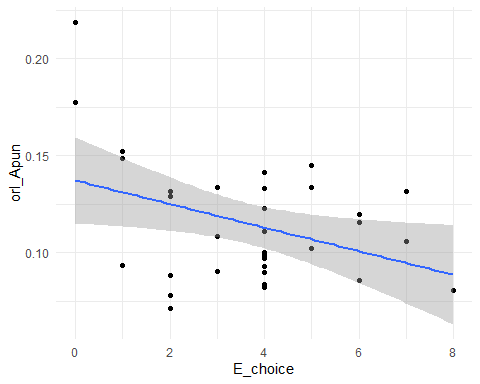
e <- ggplot(mergedPars, aes(x = E\_choice, y = orl\_Arew))  
 # Scatter plot with regression line  
 e + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



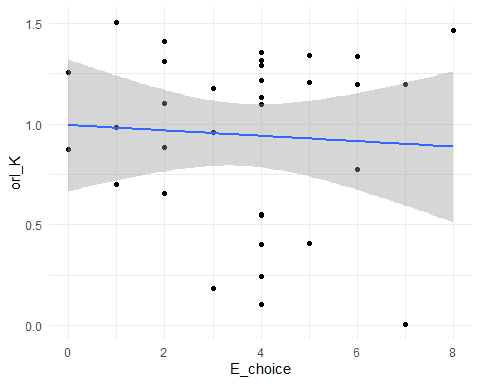
f <- ggplot(mergedPars, aes(x = E\_choice, y = orl\_Apun))  
 # Scatter plot with regression line  
 f + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



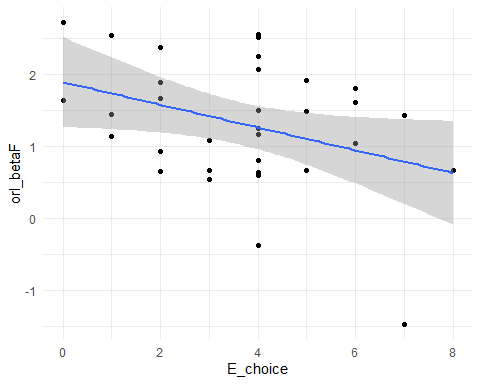
g <- ggplot(mergedPars, aes(x = E\_choice, y = orl\_K))  
 # Scatter plot with regression line  
 g + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



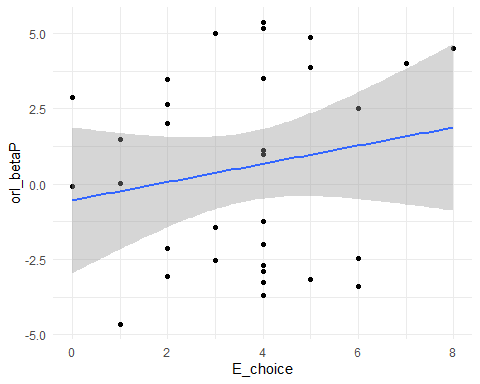
h <- ggplot(mergedPars, aes(x = E\_choice, y = orl\_betaF))  
 # Scatter plot with regression line  
 h + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



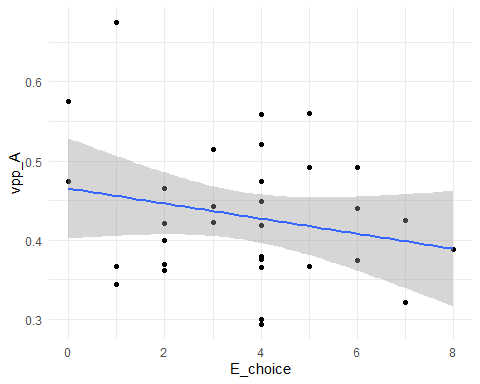
i <- ggplot(mergedPars, aes(x = E\_choice, y = orl\_betaP))  
 # Scatter plot with regression line  
 i + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



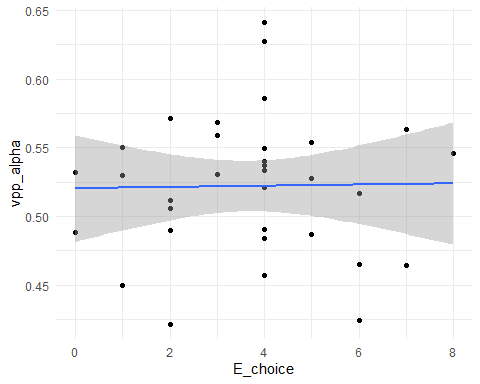
j <- ggplot(mergedPars, aes(x = E\_choice, y = vpp\_A))  
 # Scatter plot with regression line  
 j + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



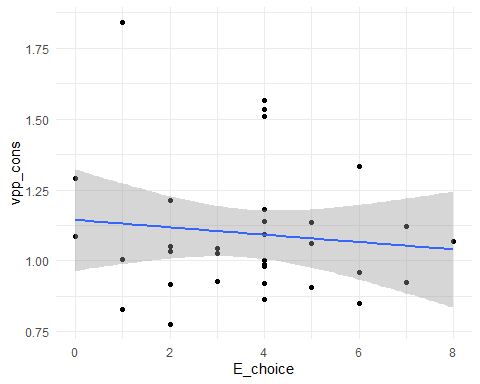
k <- ggplot(mergedPars, aes(x = E\_choice, y = vpp\_alpha))  
 # Scatter plot with regression line  
 k + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



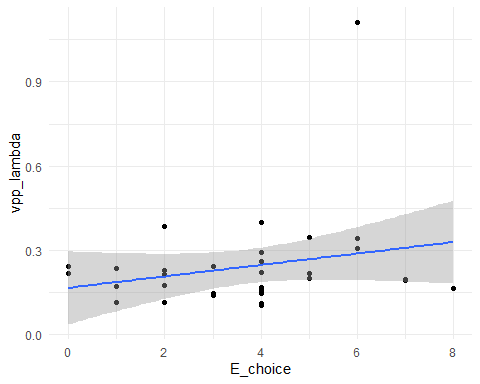
l <- ggplot(mergedPars, aes(x = E\_choice, y = vpp\_cons))  
 # Scatter plot with regression line  
 l + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



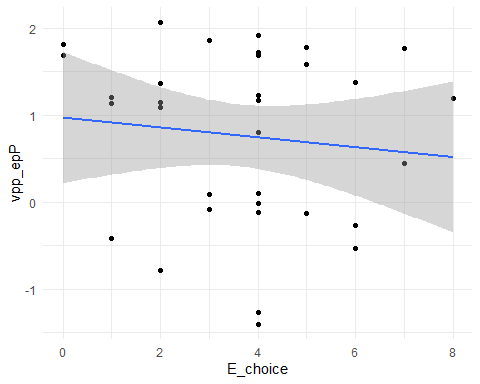
m <- ggplot(mergedPars, aes(x = E\_choice, y = vpp\_lambda))  
 # Scatter plot with regression line  
 m + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



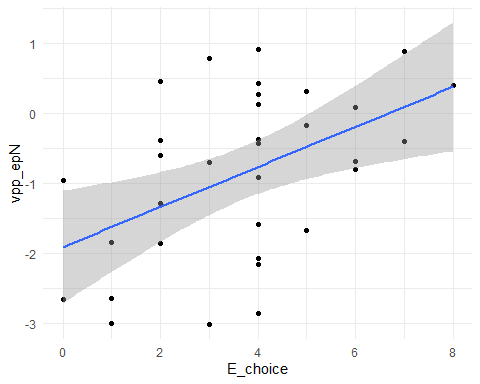
n <- ggplot(mergedPars, aes(x = E\_choice, y = vpp\_epP))  
 # Scatter plot with regression line  
 n + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



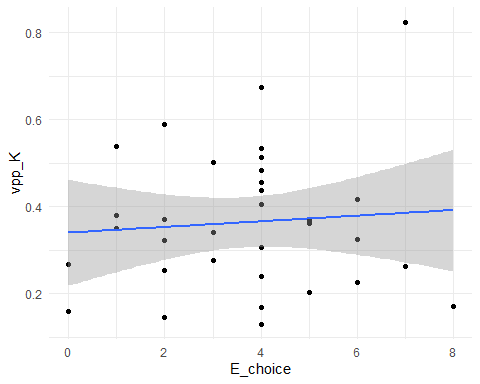
o <- ggplot(mergedPars, aes(x = E\_choice, y = vpp\_epN))  
 # Scatter plot with regression line  
 o + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



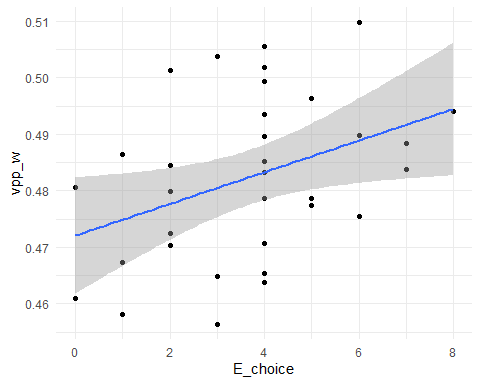
p <- ggplot(mergedPars, aes(x = E\_choice, y = vpp\_K))  
 # Scatter plot with regression line  
 p + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



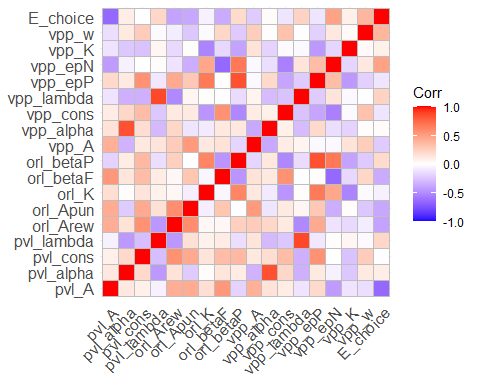
q <- ggplot(mergedPars, aes(x = E\_choice, y = vpp\_w))  
 # Scatter plot with regression line  
 q + geom\_point()+  
 geom\_smooth(method = "lm")

## `geom\_smooth()` using formula 'y ~ x'



correlation matrix

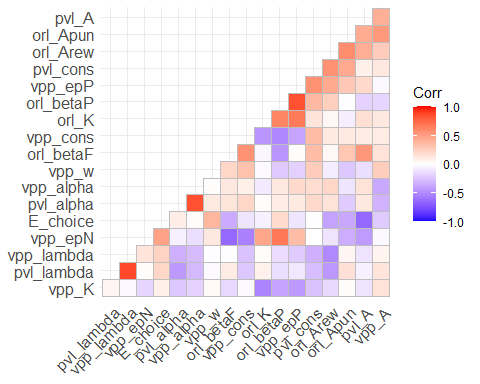
corr<-cor(mergedPars)  
ggcorrplot(corr)



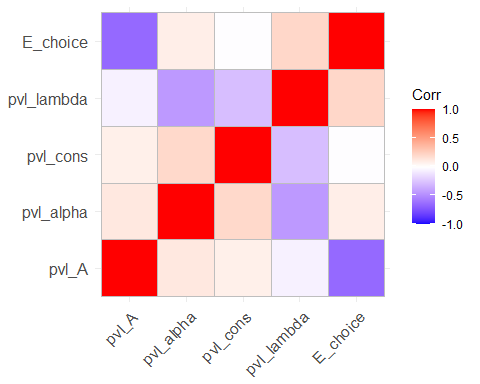
corr

## pvl\_A pvl\_alpha pvl\_cons pvl\_lambda orl\_Arew  
## pvl\_A 1.00000000 0.11929625 0.0758672890 -0.063241805 0.41820587  
## pvl\_alpha 0.11929625 1.00000000 0.2030702004 -0.438311115 0.14882302  
## pvl\_cons 0.07586729 0.20307020 1.0000000000 -0.284647548 0.55518838  
## pvl\_lambda -0.06324180 -0.43831111 -0.2846475476 1.000000000 -0.44999675  
## orl\_Arew 0.41820587 0.14882302 0.5551883815 -0.449996752 1.00000000  
## orl\_Apun 0.42878847 -0.22121579 0.4433013193 0.158292158 0.58437187  
## orl\_K 0.16822963 0.01236544 0.1393297564 0.070450706 0.04094809  
## orl\_betaF 0.52558039 0.13198846 0.3515451986 0.099838077 0.03739175  
## orl\_betaP -0.18727876 0.10314591 0.3693222808 -0.135956114 0.25487970  
## vpp\_A 0.41415162 -0.32652944 0.1220551614 0.142015778 0.26915345  
## vpp\_alpha 0.15256089 0.84086636 0.1785946915 -0.298845781 0.20500970  
## vpp\_cons 0.09754622 0.21493648 0.3478844671 -0.242900327 0.11435950  
## vpp\_lambda -0.12348071 -0.33774948 -0.3425791572 0.867041861 -0.51017958  
## vpp\_epP 0.20326412 0.11237215 0.5558680343 -0.096643240 0.43791977  
## vpp\_epN -0.42733264 -0.05663813 -0.0003022887 0.007739263 -0.12036818  
## vpp\_K -0.11890626 -0.24326260 -0.2595361882 0.049979933 -0.15661186  
## vpp\_w -0.13680009 0.11135523 0.2714780185 -0.019507255 -0.08342986  
## E\_choice -0.64823969 0.09020032 -0.0088722138 0.207925337 -0.39520870  
## orl\_Apun orl\_K orl\_betaF orl\_betaP vpp\_A  
## pvl\_A 0.428788473 0.16822963 0.52558039 -0.187278759 0.41415162  
## pvl\_alpha -0.221215785 0.01236544 0.13198846 0.103145915 -0.32652944  
## pvl\_cons 0.443301319 0.13932976 0.35154520 0.369322281 0.12205516  
## pvl\_lambda 0.158292158 0.07045071 0.09983808 -0.135956114 0.14201578  
## orl\_Arew 0.584371867 0.04094809 0.03739175 0.254879697 0.26915345  
## orl\_Apun 1.000000000 -0.06776707 0.30198423 -0.006301361 0.52279387  
## orl\_K -0.067767068 1.00000000 -0.04184297 0.615954314 0.12286233  
## orl\_betaF 0.301984235 -0.04184297 1.00000000 -0.455759237 0.15312467  
## orl\_betaP -0.006301361 0.61595431 -0.45575924 1.000000000 -0.18365051  
## vpp\_A 0.522793873 0.12286233 0.15312467 -0.183650514 1.00000000  
## vpp\_alpha -0.107737067 -0.08870474 0.12226145 0.119283073 -0.36586559  
## vpp\_cons 0.106961868 -0.46086727 0.56393454 -0.515657290 0.10166697  
## vpp\_lambda 0.047519333 0.01189878 -0.01002268 -0.150568220 0.05571982  
## vpp\_epP 0.288016908 0.66676985 0.01449111 0.839309027 -0.02797238  
## vpp\_epN -0.351373706 0.46326298 -0.65457954 0.681015747 -0.03466289  
## vpp\_K 0.012485770 -0.53403061 -0.16695435 -0.402905024 0.15140361  
## vpp\_w -0.244127144 -0.01767316 0.20786510 -0.226858954 0.25561893  
## E\_choice -0.383337226 -0.06205729 -0.35761172 0.187036939 -0.22249591  
## vpp\_alpha vpp\_cons vpp\_lambda vpp\_epP vpp\_epN  
## pvl\_A 0.152560894 0.09754622 -0.123480708 0.20326412 -0.4273326441  
## pvl\_alpha 0.840866356 0.21493648 -0.337749481 0.11237215 -0.0566381265  
## pvl\_cons 0.178594692 0.34788447 -0.342579157 0.55586803 -0.0003022887  
## pvl\_lambda -0.298845781 -0.24290033 0.867041861 -0.09664324 0.0077392627  
## orl\_Arew 0.205009704 0.11435950 -0.510179585 0.43791977 -0.1203681772  
## orl\_Apun -0.107737067 0.10696187 0.047519333 0.28801691 -0.3513737065  
## orl\_K -0.088704744 -0.46086727 0.011898782 0.66676985 0.4632629786  
## orl\_betaF 0.122261452 0.56393454 -0.010022685 0.01449111 -0.6545795387  
## orl\_betaP 0.119283073 -0.51565729 -0.150568220 0.83930903 0.6810157472  
## vpp\_A -0.365865595 0.10166697 0.055719815 -0.02797238 -0.0346628942  
## vpp\_alpha 1.000000000 0.07803172 -0.285227985 0.19830702 -0.1364660125  
## vpp\_cons 0.078031721 1.00000000 -0.262844551 -0.38819786 -0.5488482760  
## vpp\_lambda -0.285227985 -0.26284455 1.000000000 -0.22365079 0.1408164071  
## vpp\_epP 0.198307016 -0.38819786 -0.223650791 1.00000000 0.3544742115  
## vpp\_epN -0.136466013 -0.54884828 0.140816407 0.35447421 1.0000000000  
## vpp\_K -0.194839368 -0.01913956 -0.025775622 -0.44255434 -0.1787866680  
## vpp\_w 0.008472844 0.32077873 -0.009229032 -0.20232990 0.1237070739  
## E\_choice 0.017188064 -0.10993911 0.232690121 -0.11400459 0.4762686244  
## vpp\_K vpp\_w E\_choice  
## pvl\_A -0.11890626 -0.136800093 -0.648239690  
## pvl\_alpha -0.24326260 0.111355230 0.090200324  
## pvl\_cons -0.25953619 0.271478018 -0.008872214  
## pvl\_lambda 0.04997993 -0.019507255 0.207925337  
## orl\_Arew -0.15661186 -0.083429858 -0.395208701  
## orl\_Apun 0.01248577 -0.244127144 -0.383337226  
## orl\_K -0.53403061 -0.017673164 -0.062057286  
## orl\_betaF -0.16695435 0.207865104 -0.357611715  
## orl\_betaP -0.40290502 -0.226858954 0.187036939  
## vpp\_A 0.15140361 0.255618928 -0.222495906  
## vpp\_alpha -0.19483937 0.008472844 0.017188064  
## vpp\_cons -0.01913956 0.320778726 -0.109939105  
## vpp\_lambda -0.02577562 -0.009229032 0.232690121  
## vpp\_epP -0.44255434 -0.202329901 -0.114004587  
## vpp\_epN -0.17878667 0.123707074 0.476268624  
## vpp\_K 1.00000000 0.034595624 0.080553035  
## vpp\_w 0.03459562 1.000000000 0.383805836  
## E\_choice 0.08055304 0.383805836 1.000000000

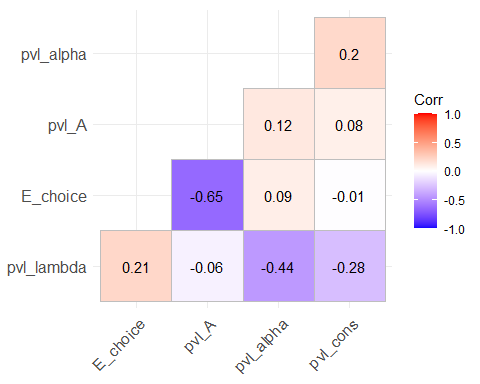
ggcorrplot(corr, hc.order = TRUE, type = "lower",  
 lab = FALSE)



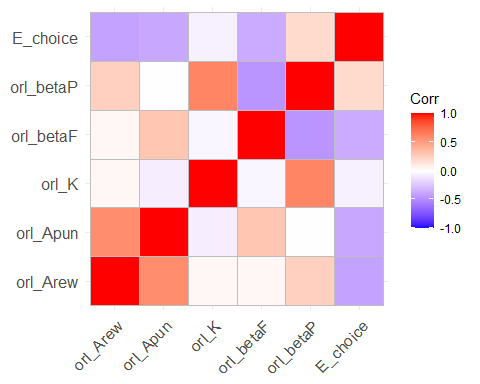
corrpvl<-cor(pvlPars)  
ggcorrplot(corrpvl)



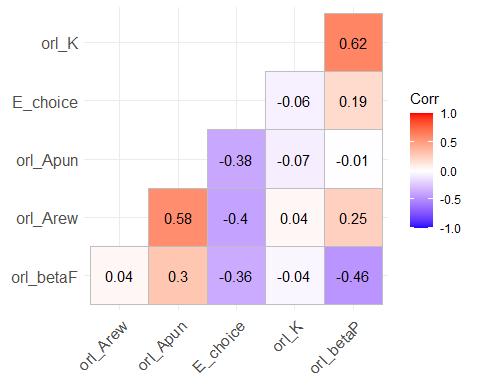
ggcorrplot(corrpvl, hc.order = TRUE, type = "lower",  
 lab = TRUE)



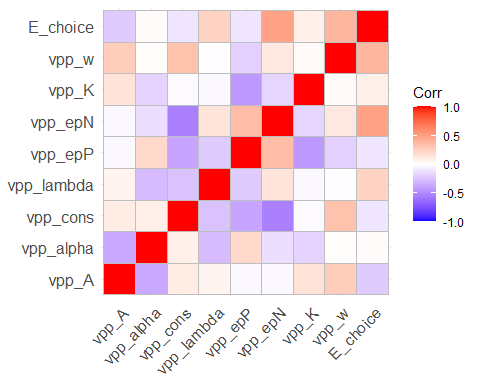
corrorl<-cor(orlPars)  
ggcorrplot(corrorl)



ggcorrplot(corrorl, hc.order = TRUE, type = "lower",  
 lab = TRUE)



corrvpp<-cor(vppPars)  
ggcorrplot(corrvpp)



ggcorrplot(corrvpp, hc.order = TRUE, type = "lower",  
 lab = TRUE)

