Dear all,

For this week's support e-mail, I would like to promote the use of correlation plots for data exploration, specifically with the R package `corrplot`. The package vignettes are an excellent reference with many variations to copy-paste. However, for inexperienced users, you may have trouble adapting `corrplot` to data sets with categorical variables, missing values, and/or 0 variances.

For this reason I wrote a wrapper script (`cpplus.r`) which handles such data automatically for quick plotting, and which also includes some helpful visual defaults (see attached example). Colors and numbers are overlaid, non-significant correlations are suppressed (at a specified alpha level), and variables can be reordered optionally with hierarchical clustering to reveal interesting variable groupings. The color scheme preserves contrast for color blindness.

The function can be run as follows:

```
# Load the cpplus package
cpplus::cpplus()
```
library(corrplot)
library(Hmisc)
library(psych)
source("C:/myfilepath/.../cpplus.r")
cpplus(mydata, reorder=TRUE, alpha=0.005)

With mydata one of your data sets that you have loaded in R (typically a data frame object). The default in cpplus is for reordering to be switched on, and for a significance level of 0.005. Of course you can always go back to the original corrplot function if specific tweaking is required. Correlation mosaics like these can be enormously helpful to make sense of association patterns. As well, they can be adapted to visualize regression coefficients, causal models, and confusion matrices based on MCC coefficients.

Best,
Ben

---

Ben Meuleman, Ph.D.
Statistician
Swiss Center for Affective Sciences
University of Geneva | Campus Biotech
Chemin des Mines 9 | CH-1202 Genève
ben.meuleman@unige.ch | +41 (0)22 379 09 79

#########################################################
## CORRPLOT PLUS FUNCTION
#########################################################

library(psych)
library(corrplot)
library(Hmisc)
dummify <- function(D) {
  as.data.frame(lapply(data.frame(D),dummy.code)) }

cpplus <- function(V,reorder=TRUE,alpha=0.005,...) {
  i <- unlist(lapply(data.frame(V),is.numeric))
  if(sum(i)==0) {
    catdat <- dummify(cbind(V)[,!i])
    alldat <- catdat
  } else {
    numdat <- cbind(V[,i])
    if(sum(i)==ncol(cbind(V))) {
      alldat <- numdat
    } else {
      
 1 Alternatively, copy the code at the bottom of this document and run it directly in R
catdat <- dummify(cbind(V)[,!i])
alldat <- data.frame(numdat,catdat)
}
alldat <- as.data.frame(alldat)
allcor <- cor(alldat,use="pairwise.complete.obs")
cat("\n",sum(is.na(allcor[lower.tri(allcor)])),"missing correlations set to zero\n\n")
flush.console()
allcor[is.na(allcor)] <- 0
diag(allcor) <- 1

pmat <- rcorr(as.matrix(alldat),type="pearson")$P ; diag(pmat) <- 1
pmat[is.na(pmat)] <- 1

allcor2 <- ifelse(pmat>alpha,0,allcor)
allcor2col <- ifelse(pmat>alpha,adjustcolor("white"),"black")
colpal <- colorRampPalette(c("darkcyan","white","burlywood"))

if(reorder==FALSE) {
  corrplot(allcor2,method="color",tl.col="black",p.mat=pmat,
sig.level=alpha,insig="pch",pch=4,pch.cex=0.6,pch.col="grey50",
addCoefasPercent=TRUE,addgrid.col="grey",col=colpal(100),addCoef.col=allcor2col,cl.pos="b")
}
if(reorder==TRUE) {
  horder <- corrplot(allcor2,method="color",col="white",order="hclust",hclust.method="ward.D")
  horder <- rownames(horder$corr)
  corrplot(allcor2,method="color",tl.col="black",p.mat=pmat,
sig.level=alpha,insig="pch",pch=4,pch.cex=0.6,pch.col="grey50",
addCoefasPercent=TRUE,addgrid.col="grey",col=colpal(100),addCoef.col=allcor2col[horder,horder],cl.pos="b",
order="hclust",hclust.method="ward.D")
}