Depression & Cognitive Impairment

Barleiris

March 12, 2019

Load data

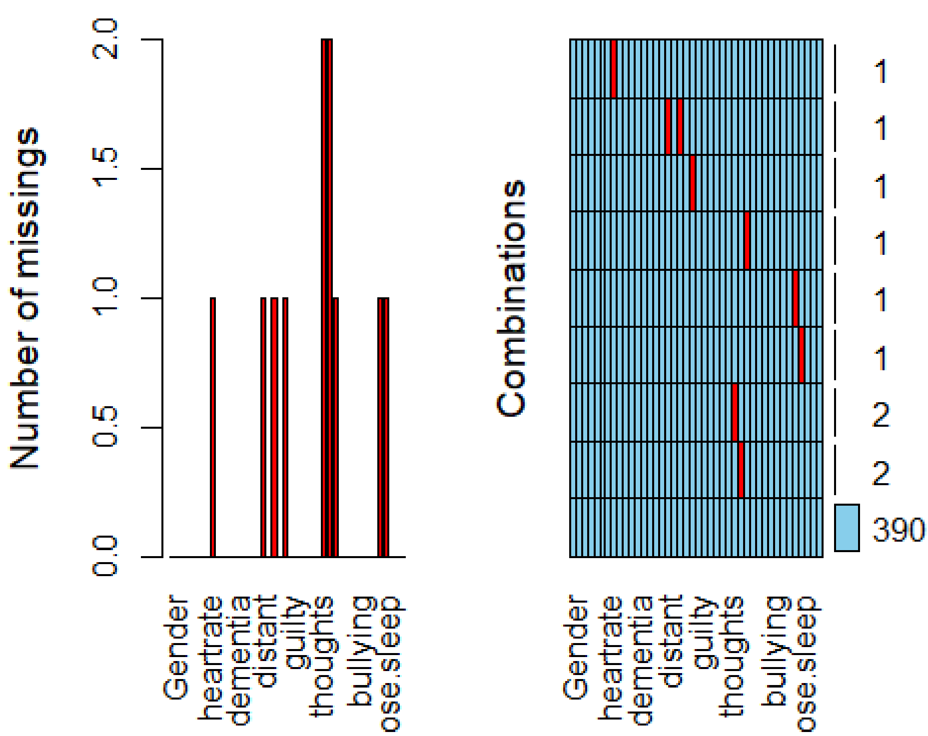
```
setwd('…')
data1 <- read.csv('….csv', sep = ',',  header = TRUE)

library(VIM)

library(Hmisc)

library(ggplot2)
library(reshape2)

library(tableone)
library(Matching)

library(ipw)

library(survey)

library(MatchIt)

library(CCA)

library(vegan)
```

Data processing

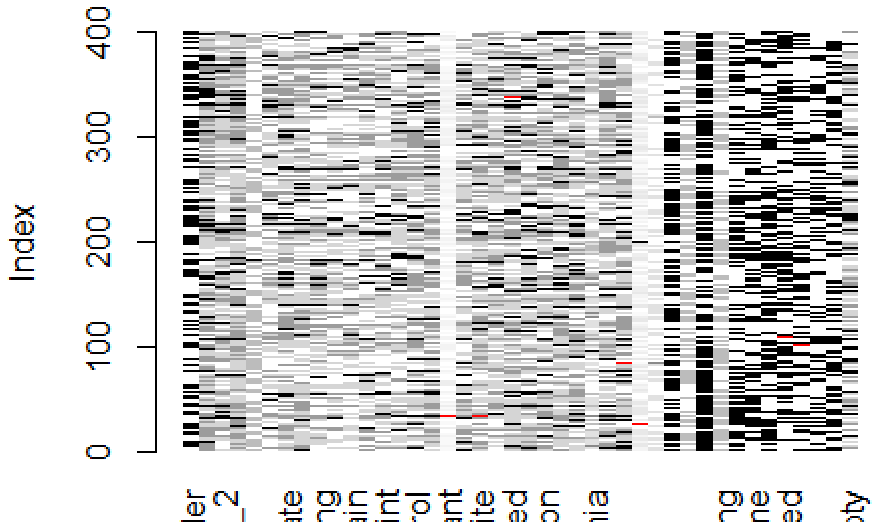
The data contains no extreme values. 9 variables out of 41 have 1 or 2 missing values, which don't have correlations with each other, hence we assume the missingness is MAR. Therefore median imputations could be used for the NAs, or we simply delete the observations with NAs. Most of the variables are positively correlated with the correlation coefficient close to zero. In the heatmap, negative correlations are blue, positive ones are red.

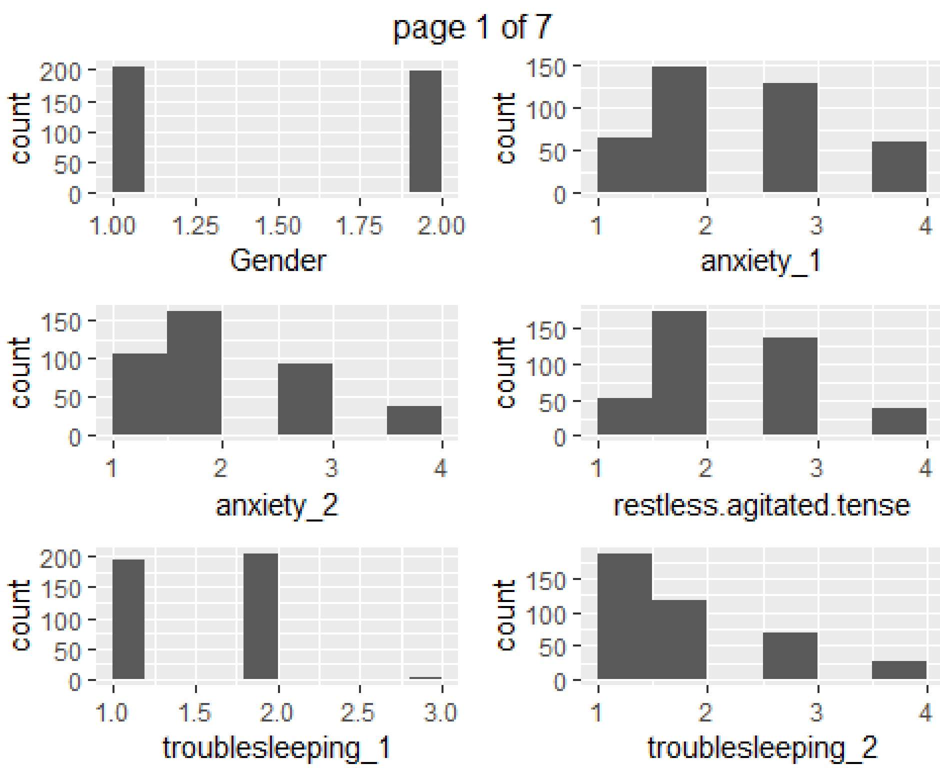
```
# descriptive analysis
summary(data1)

##      Gender        anxiety_1       anxiety_2    restless.agitated.te
nse
##  Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.   :1.000

##  1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.00   1st Qu.:2.000

##  Median :1.000   Median :2.000   Median :2.00   Median :2.000
```

```
##   Mean   :1.492   Mean   :2.455   Mean   :2.16   Mean   :2.397

##   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:3.00   3rd Qu.:3.000

##   Max.   :2.000   Max.   :4.000   Max.   :4.00   Max.   :4.000


##   Max.   :4.00   Max.   :4.000   Max.   :4.00   Max.   :4.000
##
##      distant        body.weight      appetite      slow.down
##   Min.   : 1.000   Min.   :1.000   Min.   :1.00   Min.   :1.000
##   1st Qu.: 1.000   1st Qu.:1.000   1st Qu.:1.00   1st Qu.:1.000
##   Median : 2.000   Median :2.000   Median :2.00   Median :2.000
##   Mean   : 2.175   Mean   :2.103   Mean   :2.15   Mean   :2.095
##   3rd Qu.: 3.000   3rd Qu.:3.000   3rd Qu.:3.00   3rd Qu.:3.000
##   Max.   :21.000   Max.   :4.000   Max.   :4.00   Max.   :4.000
##   NA's   :1                        NA's   :1


aggr(data1, prop = FALSE, numbers = TRUE)
```

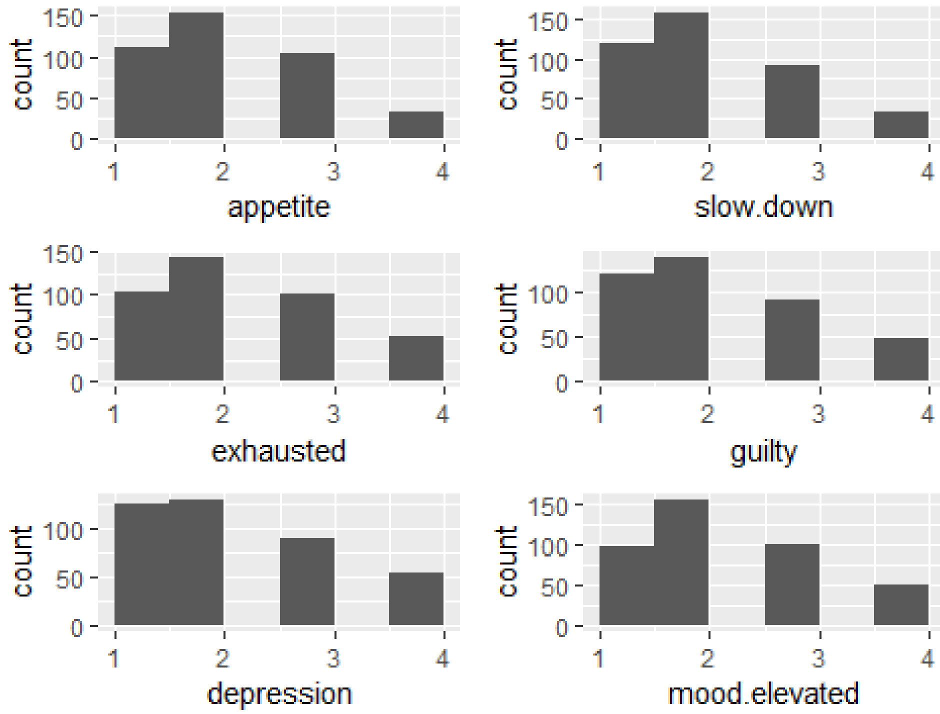



```
matrixplot(data1)
visualize(data1, vnam = TRUE)
```

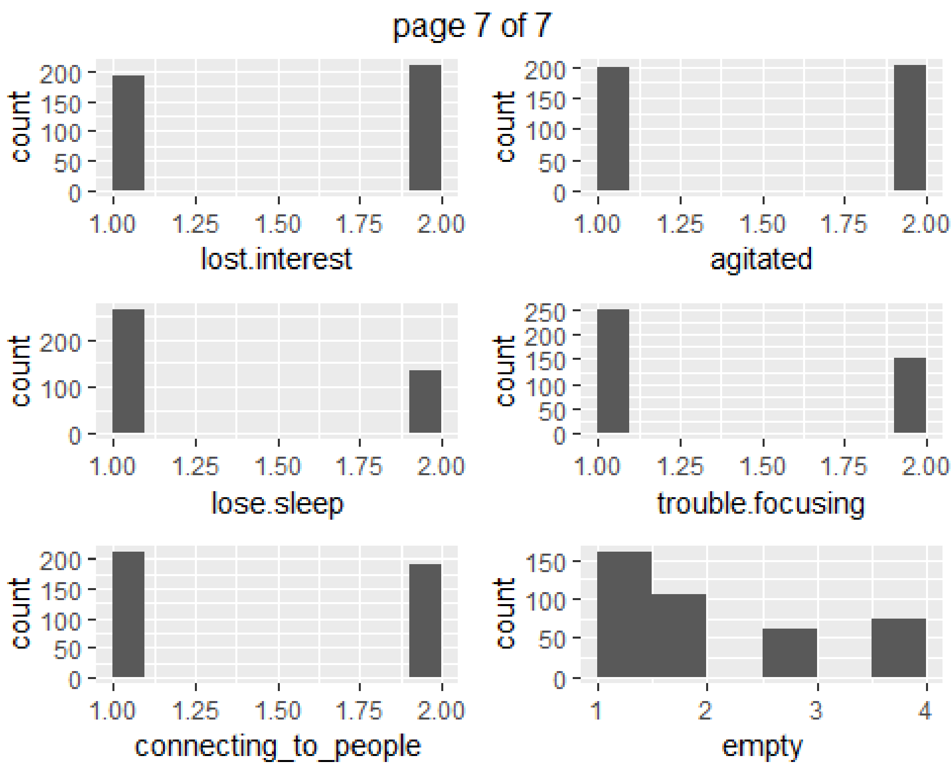

Index
0
100
200
300
400


page 1 of 7
count
Gender
anxiety_1
anxiety_2
restless.agitated.tense
troublesleeping_1
troublesleeping_2

page 4 of 7

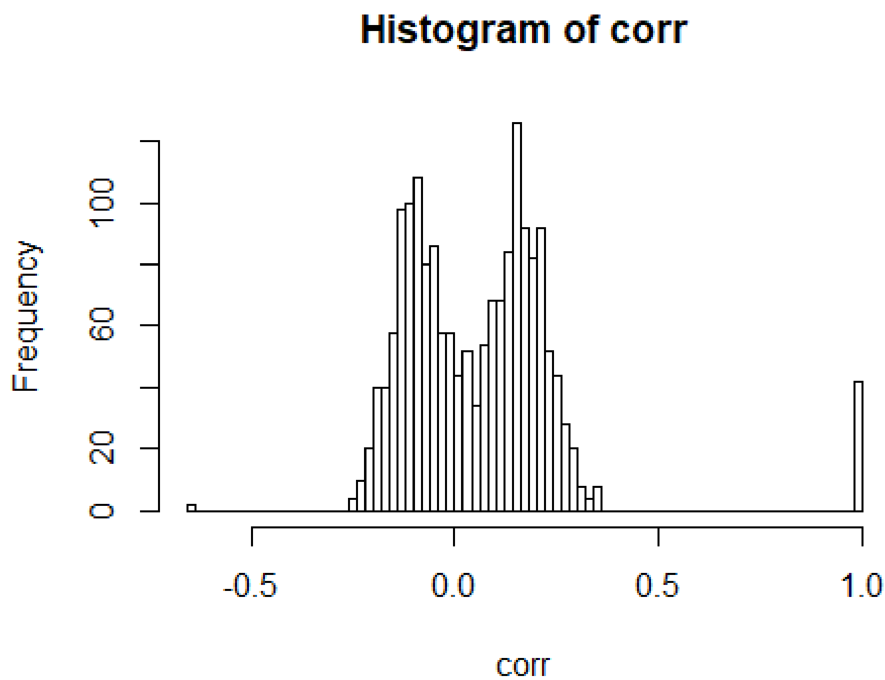


page 5 of 7

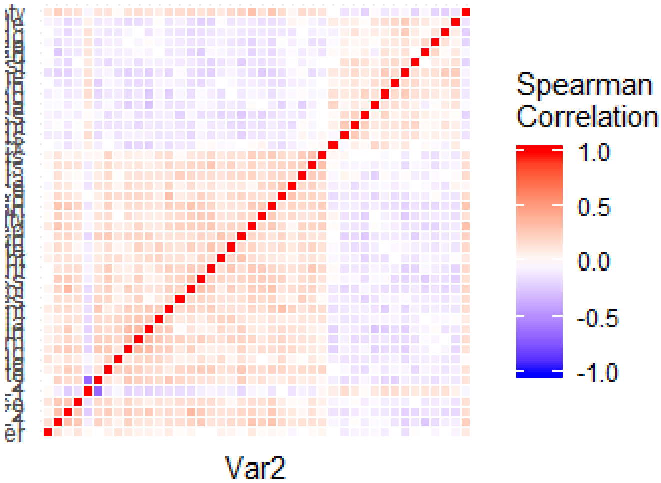



```
# missing data imputation
for(i in 1:length(data1)){
  data1[,i] <- impute(data1[,i], median)
  data1[,i] <- as.numeric(data1[,i])
}
matrixplot(data1)
```

```
# correlation
corr <- round(cor(data1, method = 'spearman'),2)
hist(corr, breaks = 100)
```




```
heatMap <- function(data, color_limit, color_name){
  corr <- melt(data)
ggplot(corr, aes(Var2, Var1, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpo
int = 0, limit = color_limit, space = "Lab", name = color_name) +
  theme_minimal() +
  theme(axis.text.x = element_blank()) +
  coord_fixed()
}

heatMap(corr, color_limit = c(-1,1), color_name = "Spearman\nCorrelatio
n")
```




```
# classify questions
confounding <- data1[,c(1,7,8,9,11,20,28,31)]
depression <- data1[,c(2,3,4,5,6,15,16,17,18,19,21,22,23,29,30,32,33,34,
35,36,37,39,41,42)]
cognitive <- data1[,c(10,12,13,14,24,25,26,27,38,40)]
```

Inscidence of cooccurence

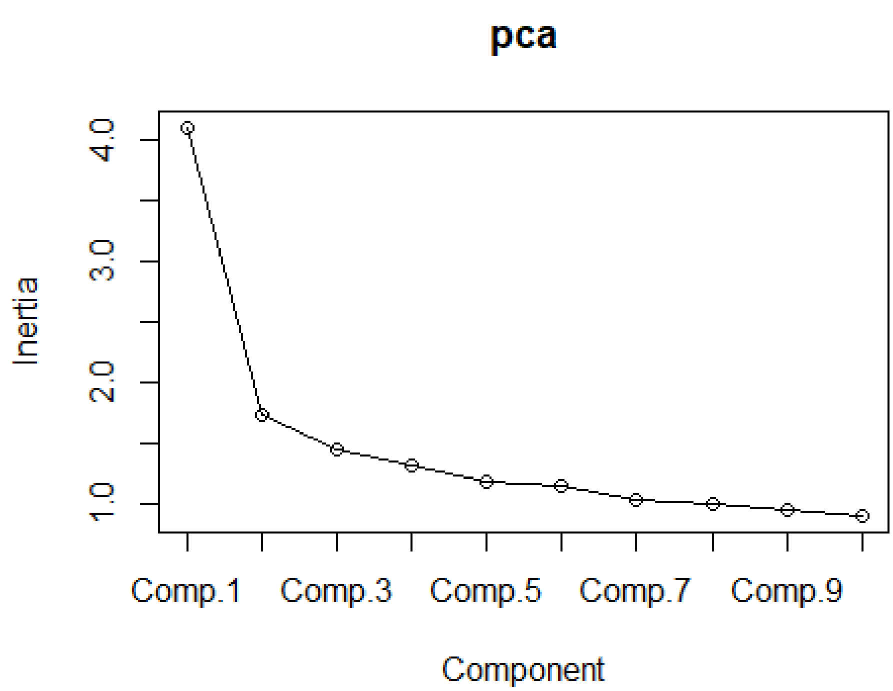
kmeans has a drawback that we cannot explain why a subject is classified into one group instead of another. Pca has a problem that no principle component can be extracted. Hence maybe we will use anormally detection to find the subjects with a risk of getting depression or cognitive impairment, and then calculate the possibility of cooccurence conditioned on two cases respectively. It seems the occurence of depression and cognitive impairment is more common in the cognitive impaired group than in the depressed group.

```
fit <- kmeans(depression, 5)
fit
## 5      1.591304   1.382609             1.626087 1.713043
##
## Clustering vector:
##   [1] 1 4 2 2 4 5 5 2 5 5 2 5 4 2 5 5 1 5 5 2 2 3 5 1 3 2 3 1 2 5 1
3 3 2 2

##
## Within cluster sum of squares by cluster:
```

```
## [1]  922.6707  993.6351 1215.5930  902.7442 1586.8696
##  (between_SS / total_SS =  25.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

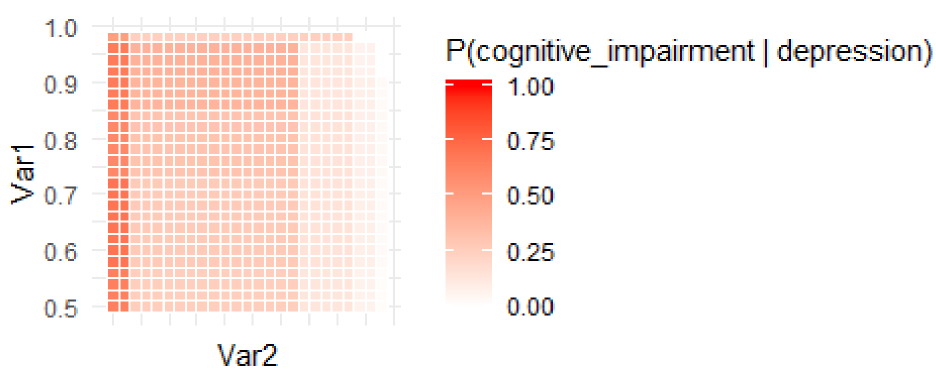
```
pca <- princomp(depression, cor = TRUE)
summary(pca, loading = TRUE)
```

```
## Importance of components:
##                           Comp.1     Comp.2     Comp.3     Comp.4
## Standard deviation     2.0242141 1.31127048 1.19966537 1.14153119
## Proportion of Variance 0.1707268 0.07164293 0.05996654 0.05429556
## Cumulative Proportion  0.1707268 0.24236971 0.30233625 0.35663181
##                            Comp.5     Comp.6     Comp.7    Comp.8
## Standard deviation     1.08436590 1.06640697 1.01096203 0.9982408
## Proportion of Variance 0.04899372 0.04738433 0.04258518 0.0415202
## Cumulative Proportion  0.40562554 0.45300986 0.49559504 0.5371152
##                            Comp.9    Comp.10    Comp.11    Comp.12sc
reeplot(pca, type = 'lines')
```

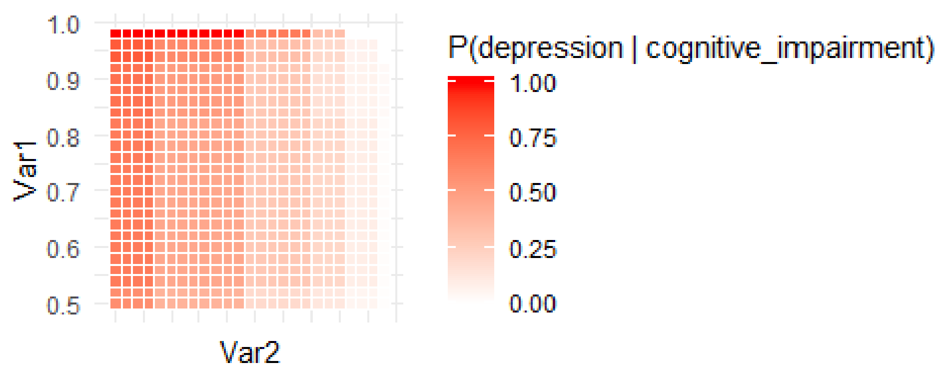



```
findAtRisk <- function(data, quantil){
  for (i in 1:length(data)) {
    data[,i] <- ifelse(data[,i] == max(data[,i]),1,0)
```

```
    }
  risk_score <- rowMeans(data)
  at_risk <- which(risk_score > quantile(risk_score, quantil))
  return(at_risk)
}

coOccurence <- function(data1, data2, quantil){
  #the possibility of data2 conditioned on data1
  breaks <- length(quantil)
  cooccur_matrix <- matrix(data = NA, nrow = breaks, ncol = breaks)
  for(i in 1:breaks){
    for(j in 1:breaks){
      array1 <- findAtRisk(data1, quantil[i])
      array2 <- findAtRisk(data2, quantil[j])
      cooccur_inscidence <- intersect(array1, array2)
      cooccur_matrix[i,j] <- length(cooccur_inscidence)/length(array1)
    }
  }
  row.names(cooccur_matrix) <- as.character(quantil)
  colnames(cooccur_matrix) <- as.character(quantil)
  return(cooccur_matrix)
}

quantil <- seq(0.5, 0.99, 0.02)
dep2cog <- coOccurence(depression, cognitive, quantil)
cog2dep <- coOccurence(cognitive, depression, quantil)

par(mfrow = c(1,2))
heatMap(dep2cog, color_limit = c(0,1), color_name = "P(cognitive_impair
ment | depression)")
```



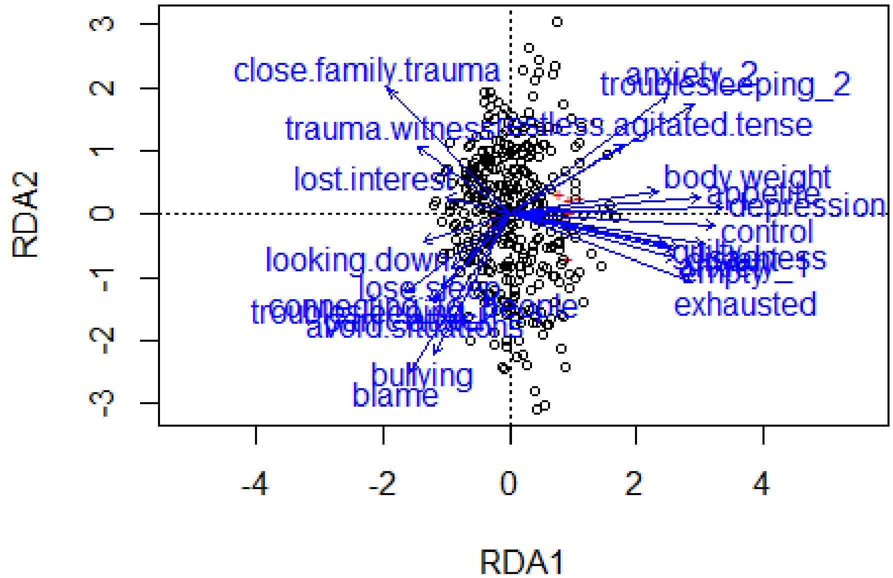
```r
heatMap(cog2dep, color_limit = c(0,1), color_name = "P(depression | cog
nitive_impairment)")
```



##

Redundancy Analysis

```r
decorana(cognitive)

##
## Call:
## decorana(veg = cognitive)
##
## Detrended correspondence analysis with 26 segments.
## Rescaling of axes with 4 iterations.
##
##                     DCA1    DCA2    DCA3    DCA4
## Eigenvalues      0.02527 0.02309 0.02020 0.01732
## Decorana values  0.02528 0.02261 0.01947 0.01604
## Axis lengths     0.79095 0.74932 0.70305 0.68282

sp0 <- rda(cognitive ~., depression)
sp0


## 1.3569 0.8961 0.8519 0.7090 0.6611 0.6407 0.5585 0.4889 0.2098 0.182
6

plot(sp0)
```

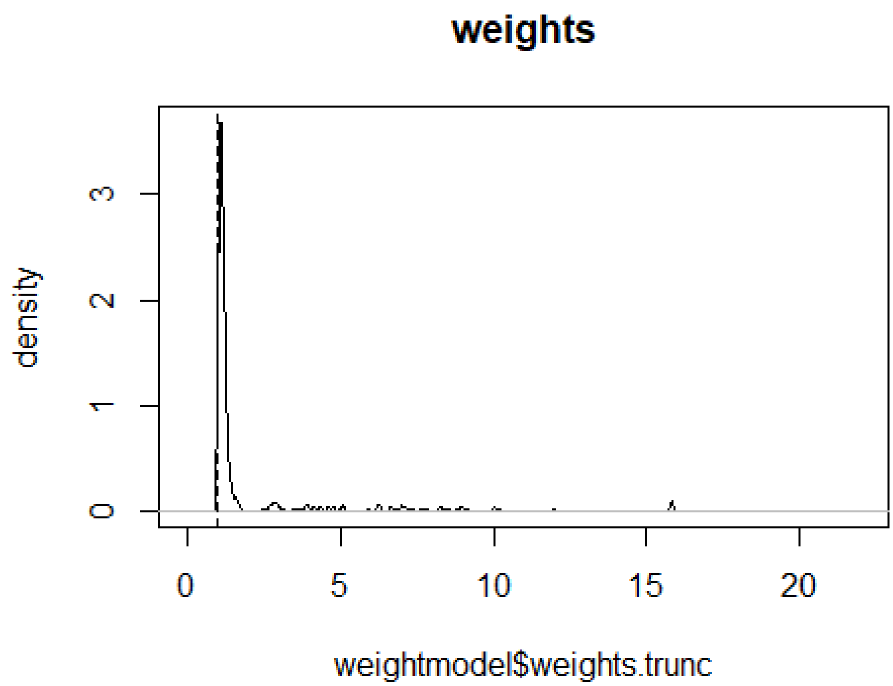


Causal Inference

Next we use IPSW to compare the score of cognitive impairment in people who feel depressed and who don't. The 95% confidence interval of depression's effect on cognition does not contain 0, which means the relationship between depression and cognitive impairment is statistically significant. However, cognition does not have a significant correlation to depression after adjusting for confoundings.

```
causalEffect <- function(treat_data, confounding, outcome_data, quantil)
{

  print(paste('Effect of', deparse(substitute(treat_data)), 'on', depar
se(substitute(outcome_data))))

  treated <- findAtRisk(treat_data, quantil)
  treat <- rep(0, length(data1[,1]))
  treat[treated] <- 1

  outcome <- rowMeans(outcome_data)

  xvars <- colnames(confounding)

  data3 <- cbind(confounding,treat)

  # Propensity score matching
  psmodel <- glm(treat ~ ., family = binomial(link = 'logit'), data = d
```

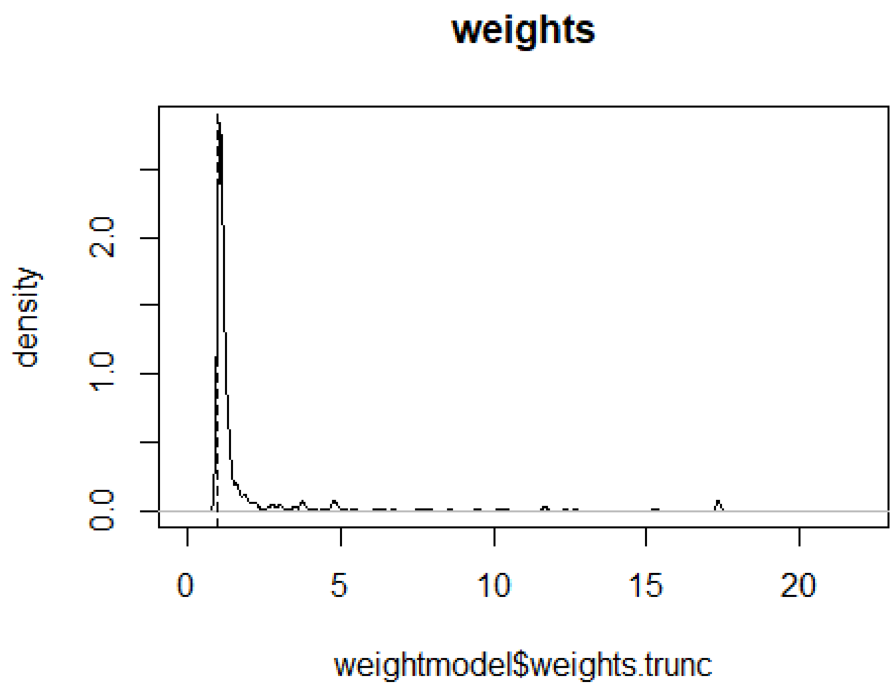
```
ata3)
  ps <- predict(psmodel, type = 'response')
  weight <- ifelse(data3$treat == 1, 1/(ps), 1/(1-ps))
  weighteddata <- svydesign(ids = ~1, data =data3, weights = ~ weight)

  weightedtable <- svyCreateTableOne(vars = xvars, strata = 'treat', da
ta =weighteddata, test = FALSE)
  print(weightedtable, smd = TRUE)

  # 95% C.I. for causal effect
  weightmodel <- ipwpoint(exposure = treat, family = 'binomial', link =
 'logit',
                           denominator = ~ ., data = data3, trunc = .0
1)
  summary(weightmodel$weights.trunc)

  ipwplot(weights = weightmodel$weights.trunc, logscale = FALSE, main =
 'weights', xlim = c(0,22))
  data3$wt <- weightmodel$weights.trunc

  data3 <- cbind(outcome, data3)
  msm <- (svyglm(outcome ~ treat, design = svydesign(~1, weights = ~ wt,
 data = data3)))
  print(coef(msm))
  print(confint(msm))
}

causalEffect(depression, confounding, cognitive, 0.8)

## [1] "Effect of depression on cognitive"
##                                 Stratified by treat
##                                  0              1              SMD
##   n                              399.62         417.31
##   Gender (mean (sd))               1.49 (0.50)    1.46 (0.50)  0.068
##   te (mean (sd))                   2.17 (0.95)    2.05 (1.00)  0.116
##   ng (mean (sd))                   2.11 (0.94)    1.99 (0.80)  0.138
##   ng (mean (sd))                   1.91 (0.98)    1.89 (0.91)  0.026
##   chest.pain (mean (sd))           1.83 (0.97)    1.75 (0.91)  0.082
##   n (mean (sd))                    2.09 (0.90)    1.99 (1.05)  0.108
##   mu (mean (sd))                   2.33 (0.99)    2.23 (1.04)  0.096
##   tt (mean (sd))                   1.59 (0.49)    1.52 (0.50)  0.145
```




```
## (Intercept)       treat
##   1.9355813   0.1558179
##                   2.5 %    97.5 %
## (Intercept) 1.89302534 1.9781373
## treat       0.03161911 0.2800167
```

```
causalEffect(cognitive, confounding, depression, 0.8)
```

```
## [1] "Effect of cognitive on depression"
##                              Stratified by treat
##                               0             1             SMD
##   n                           398.05        433.03
##   Gender (mean (sd))            1.49 (0.50)   1.56 (0.50)  0.132
##   te (mean (sd))                2.16 (0.93)   2.07 (1.05)  0.097
##   ng (mean (sd))                2.10 (0.91)   2.08 (1.05)  0.022
##   ng (mean (sd))                1.90 (0.95)   1.82 (0.96)  0.086
##   cn (mean (sd))                1.83 (0.96)   1.85 (1.00)  0.029
##   sn (mean (sd))                2.10 (0.88)   2.14 (1.13)  0.039
##   ls (mean (sd))                2.32 (0.96)   2.14 (1.16)  0.163
##   tt (mean (sd))                1.59 (0.49)   1.54 (0.50)  0.093
```




```
## (Intercept)       treat
##  1.93062460  0.04887152
##                   2.5 %    97.5 %
## (Intercept)  1.90344194 1.9578073
## treat       -0.03214008 0.1298831
```