eda

Jared Gage

4/24/2021

```r
id <- 1:15051

keep <- de[, 1:15] # Question 1:10

nmu <- cbind(id, keep, de[, str_detect(names(de), "NMU")]) # Keep all non-medical use
use <- cbind(id, keep, de[, str_detect(names(de), "_USE")]) # Keep all non-medical use


pain <- cbind(id, keep, de[, 16:37]) # Question 11:32
stim <- cbind(id, keep, de[, 38:39]) # Question 33, 34
benz <- cbind(id, keep, de[, 40:41]) # 35, 36
thc <- cbind(id, keep, de[, 42:43]) # 37, 38

end_of_survey <- cbind(id, keep, de[, 44:61]) # Questions 39:42

help <- cbind(id, keep, de[, 62:70]) # Question 43:45

pain_chronic <- cbind(id, keep, de[, 71:76]) # Question 46:48

rx_drug_use <- cbind(id, keep, de[, 77:83]) # Question 52:56

mental <- cbind(id, keep, de[, 84:97]) # Question 57

dast_10 <- cbind(id, keep, de[, 98:107]) # Question 58:67

summary <- cbind(id, keep, de[, 108:156]) # Question 58:67
```

```r
write.csv(nmu, "nmu.csv", row.names = FALSE)
write.csv(use, "use.csv", row.names = FALSE)
write.csv(pain, "pain.csv", row.names = FALSE)
write.csv(stim, "stim.csv", row.names = FALSE)
write.csv(benz, "benz.csv", row.names = FALSE)
write.csv(thc, "thc.csv", row.names = FALSE)
write.csv(end_of_survey, "end_of_survey.csv", row.names = FALSE)
write.csv(help, "help.csv", row.names = FALSE)
write.csv(pain_chronic, "pain_chronic.csv", row.names = FALSE)
write.csv(rx_drug_use, "rx_drug_use.csv", row.names = FALSE)
write.csv(mental, "mental.csv", row.names = FALSE)
write.csv(dast_10, "dast_10.csv", row.names = FALSE)
write.csv(summary, "summary.csv", row.names = FALSE)
```

```r
#install.packages("reshape2")
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.0.5
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.4
```

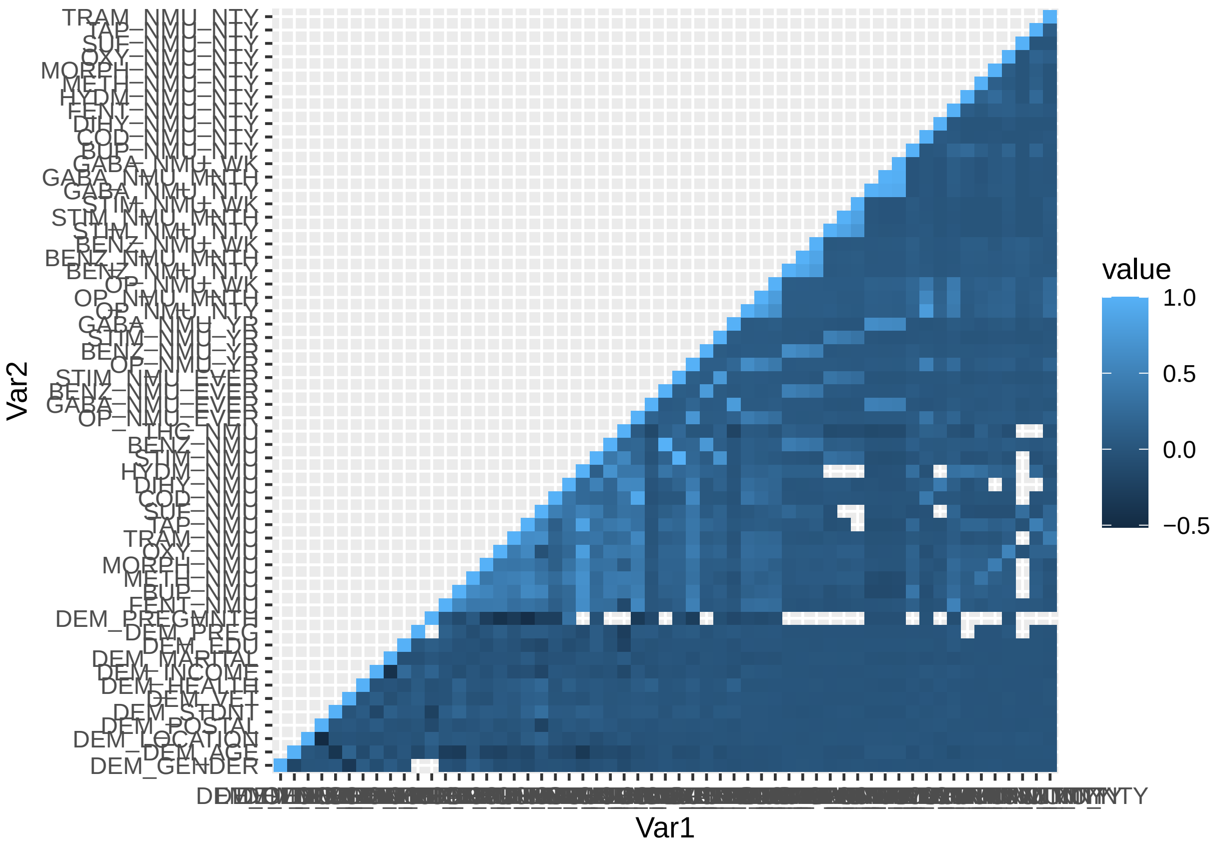
```r
 get_lower_tri<-function(cormat){
    cormat[upper.tri(cormat)] <- NA
    return(cormat)
 }

reorder_cormat <- function(cormat){
# Use correlation between variables as distance
dd <- as.dist((1-cormat)/2)
hc <- hclust(dd)
cormat <-cormat[hc$order, hc$order]
}
```

```r
## analysis of demographics vs nmu
nmu <- cbind(id, keep, de[, str_detect(names(de), "NMU")])
nmu <- nmu[, 5:61]
df <- round(cor(nmu, use="pairwise.complete.obs"), 4)
```

```
## Warning in cor(nmu, use = "pairwise.complete.obs"): the standard deviation is
## zero
```

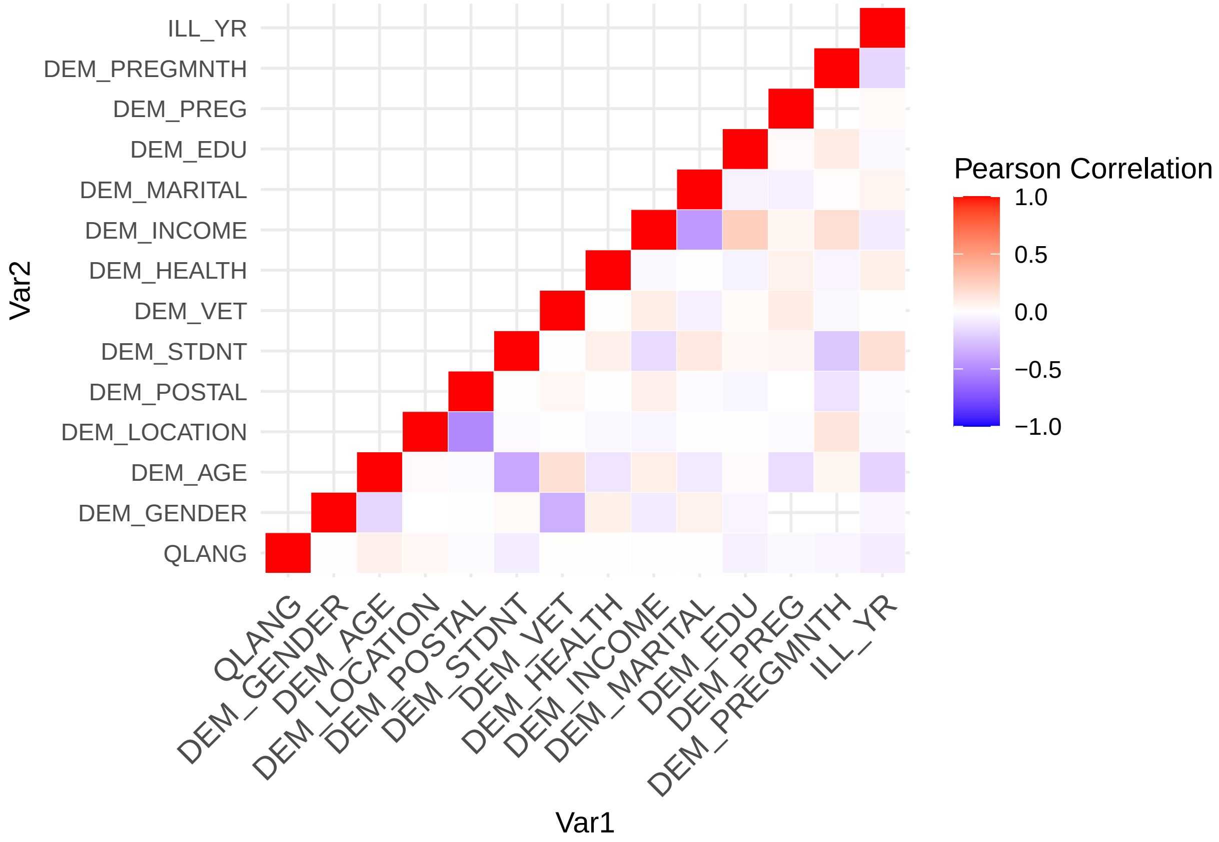
```r
df <- get_lower_tri(df)
meltedData <- melt(df, na.rm = TRUE)
#meltedData <- meltedData[complete.cases(meltedData),]
ggplot(data = meltedData, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()
```



```r
newdf <- cbind(keep[3:15], summary[52])
user_dem_df <- round(cor(newdf, use="pairwise.complete.obs"), 4)
```

```
## Warning in cor(newdf, use = "pairwise.complete.obs"): the standard deviation is
## zero
```

```r
user_dem_df <- get_lower_tri(user_dem_df)
meltedData2 <- melt(user_dem_df, na.rm = TRUE)
#meltedData <- meltedData[complete.cases(meltedData),]
ggplot(data = meltedData2, aes(x=Var1, y=Var2, fill=value)) +
   geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
    name="Pearson Correlation") +
  theme_minimal()+ # minimal theme
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))
```



```r
#guiltdrug <- cbind(summary[52], dast_10[21])
cor.test(summary[, 52], dast_10[, 21] )
```

```
##
##  Pearson's product-moment correlation
##
## data:  summary[, 52] and dast_10[, 21]
## t = 10.541, df = 15049, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.06973263 0.10145076
## sample estimates:
##        cor
## 0.08561338
```

```r
mean(help[(summary[,52] == 1),17])
```

```
## [1] 0.08115672
```

```r
mean(summary[help[,17] == 1, 52])
```

```
## [1] 0.2939189
```

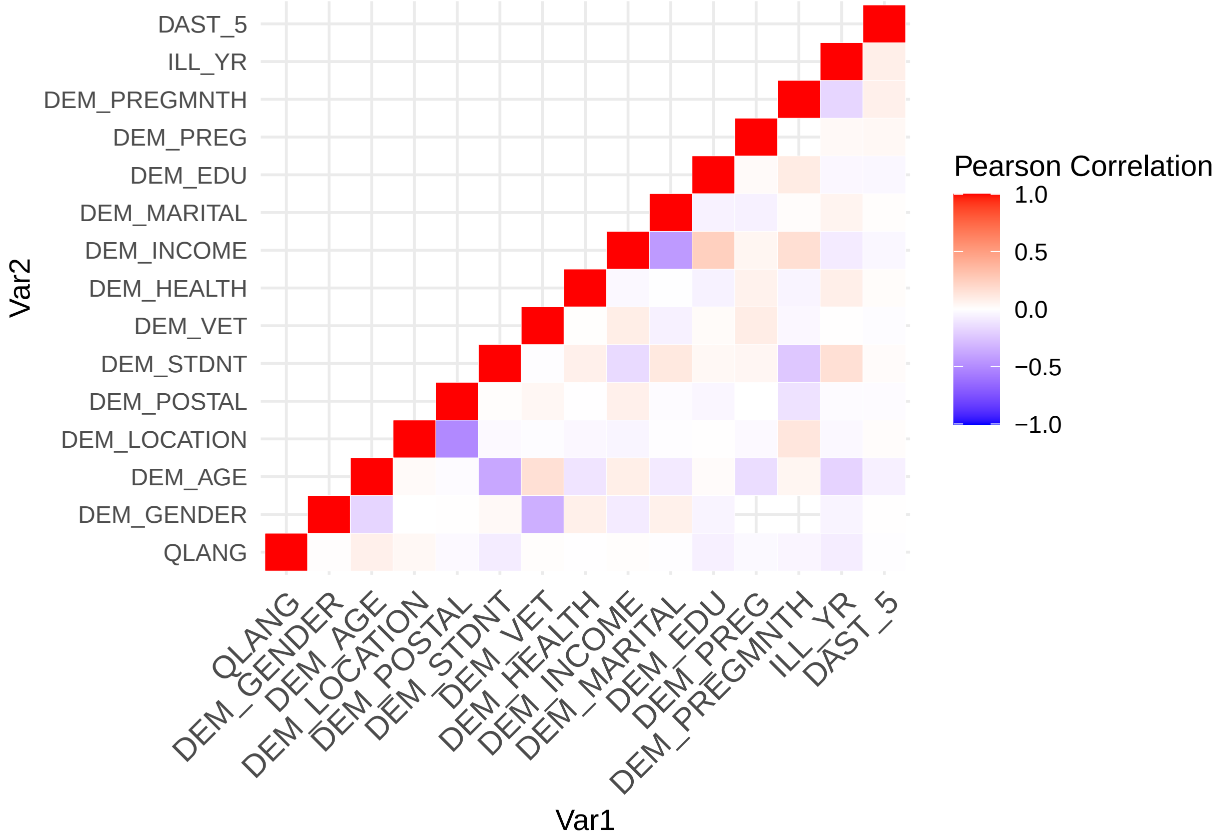
```r
cor(summary[,52], help[,17])
```

```
## [1] 0.1226353
```

```r
guiltdem <- cbind(keep[3:15], summary[52], dast_10[21])
guilt_dem_df <- round(cor(guiltdem, use="pairwise.complete.obs"), 4)
```

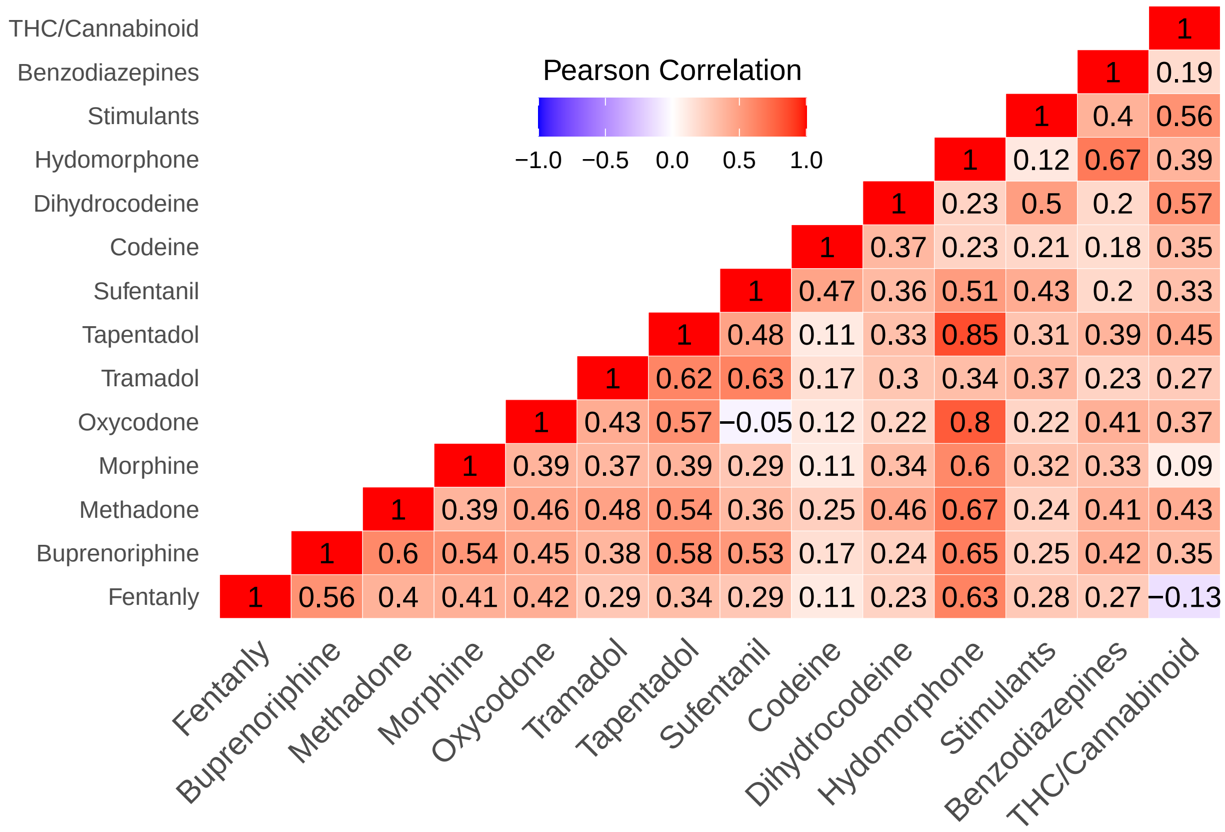
```
## Warning in cor(guiltdem, use = "pairwise.complete.obs"): the standard deviation
## is zero
```

```
guilt_dem_df <- get_lower_tri(guilt_dem_df)
meltedData3 <- melt(guilt_dem_df, na.rm = TRUE)
#meltedData <- meltedData[complete.cases(meltedData),]
ggplot(data = meltedData3, aes(x=Var1, y=Var2, fill=value)) +
   geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
    name="Pearson Correlation") +
  theme_minimal()+ # minimal theme
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))
```



```
drugs <- nmu[,13:26]
drugsdf <- round(cor(drugs, use="pairwise.complete.obs"), 2)
drugsdf <- get_lower_tri(drugsdf)
meltedData4 <- melt(drugsdf, na.rm = TRUE)
#meltedData <- meltedData[complete.cases(meltedData),]
ggheatmap <- ggplot(data = meltedData4, aes(x=Var1, y=Var2, fill=value)) +
   geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
    name="Pearson Correlation") +
  theme_minimal()+ # minimal theme
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))+
  scale_x_discrete(labels = c('Fentanly','Buprenoriphine', 'Methadone', 'Morphine',
                              'Oxycodone', 'Tramadol', 'Tapentadol', 'Sufentanil', 'Codeine',
                              'Dihydrocodeine', 'Hydomorphone', 'Stimulants', 'Benzodiazepines',
                              'THC/Cannabinoid'))+
  scale_y_discrete(labels = c('Fentanly','Buprenoriphine', 'Methadone', 'Morphine',
                              'Oxycodone', 'Tramadol', 'Tapentadol', 'Sufentanil', 'Codeine',
                              'Dihydrocodeine', 'Hydomorphone', 'Stimulants', 'Benzodiazepines',
                              'THC/Cannabinoid'))


ggheatmap +
geom_text(aes(Var1, Var2, label = value), color = "black", size = 4) +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
              title.position = "top", title.hjust = 0.5))
```



```
tiler <- function(x, columns = c(2, 3), num_col = 2L) {
  # Given that column A is 1, what is the average of Column B
  if (num_col < 2) {
    stop("Columns must be an integer value of at least 2.")
  }
  if (num_col != length(columns)) {
    stop("Number of columns must equal the length of the 'columns' vector you inputted.")
  }
  if (num_col == 2) { # Just inputting a n x 2 matrix
    # Get columns where column A == 1
    temp <- x[!is.na(x[[columns[1]]]) & x[[columns[1]]] == 1, columns]
    final <- mean(temp[, 2], na.rm = TRUE)
  } else {
    A <- matrix(rep(1, times = (num_col)^2), nrow = num_col)
    B <- permut(seq_len(num_col),2)
    for (i in seq_len((num_col)^2 - num_col)) { # This tells us how many permutations we can have
      # Formula for unique permutations is n^2
        temp <- x[!is.na(x[[columns[B[i, 1]]]]) & x[[columns[B[i, 2]]]] == 1, columns]
        value <- mean(temp[, 2], na.rm = TRUE)
        A[B[i, 1], B[i, 2]] <- value
        final <- A
    }
  }
  final
}
```

```
library(randtests)
```

```
## Warning: package 'randtests' was built under R version 4.0.3
```

```
#mydata <- tiler(drugs, columns = c(1:14), num_col = 14L)
#mydata <- as.data.frame(mydata)
#colnames(mydata) <- colnames(drugsdf)
#rownames(mydata) <- rownames(drugsdf)
#thedats <- get_lower_tri(mydata)
#meltedData5 <- melt(thedats, na.rm = TRUE)

#ggplot(data = meltedData5, aes(x=Var1, y=Var2, fill=value))+
#  geom_tile(color = "white")+
# scale_fill_gradient2(low = "blue", high = "red", mid = "white",
#   midpoint = 0, limit = c(-1,1), space = "Lab",
#    name="Pearson Correlation") +
#  theme_minimal()+ # minimal theme
# theme(axis.text.x = element_text(angle = 45, vjust = 1,
#    size = 12, hjust = 1))+
#  scale_x_discrete(labels = c('Fentanly','Buprenoriphine', 'Methadone', 'Morphine',
#                              'Oxycodone', 'Tramadol', 'Tapentadol', 'Sufentanil', 'Codeine',
#                              'Dihydrocodeine', 'Hydomorphone', 'Stimulants', #'Benzodiazepines',
#                              'THC/Cannabinoid'))+
#  scale_y_discrete(labels = c('Fentanly','Buprenoriphine', 'Methadone', 'Morphine',
#                              'Oxycodone', 'Tramadol', 'Tapentadol', 'Sufentanil', 'Codeine',
#                              'Dihydrocodeine', 'Hydomorphone', 'Stimulants', #'Benzodiazepines',
#                              'THC/Cannabinoid'))
```