Machine Learning Final

Megan Marziali

4/8/2021

# Part 1: Dietary patterns using unsupervised analysis

## Question 1: Construct a research question

Research question: Do dietary patterns during pregnancy predict postpartum mental health symptoms?

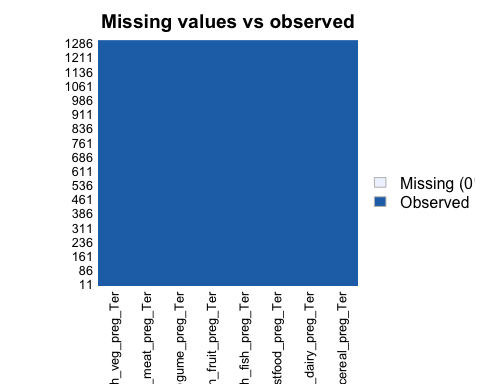
The goal of this unsupervised analysis will be to develop classes of dietary patterns among women during pregnancy. These classes will be used as an exposure in future analyses, aiming to predict whether dietary patterns during pregnancy are predictive of postpartum mental health symptoms (such as postpartum depression). This is a predictive research question.

## Question 2: Running appropriate unsupervised analysis

diet.data =   
 read.csv("./data/diet\_data.csv", header = TRUE) %>%   
 select(  
 "h\_cereal\_preg\_Ter",  
 "h\_dairy\_preg\_Ter",  
 "h\_fastfood\_preg\_Ter",  
 "h\_fish\_preg\_Ter",  
 "h\_fruit\_preg\_Ter",  
 "h\_legume\_preg\_Ter",  
 "h\_meat\_preg\_Ter",  
 "h\_veg\_preg\_Ter",  
 )  
  
# Scaling data  
colMeans(diet.data, na.rm = TRUE)  
apply(diet.data, 2, sd, na.rm = TRUE)  
  
scale(diet.data, center = TRUE, scale = TRUE)

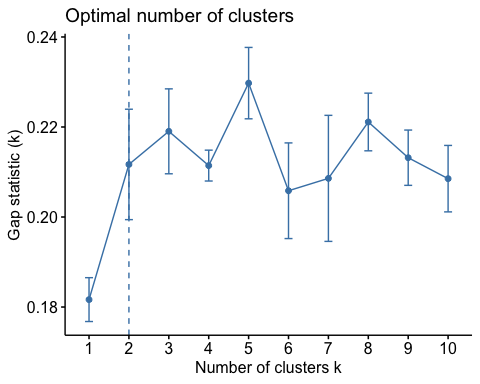
I restricted the data to variables which assessed dietary patterns during pregnancy. The total dataset is comprised of 1301 observations, and 8 variables.

missmap(diet.data, main = "Missing values vs observed")



I used missmap to determine if there are any missing observations, which there are not. To determine clusters within the data, a hierarchical cluster analysis with the complete linkage method will be carried out.

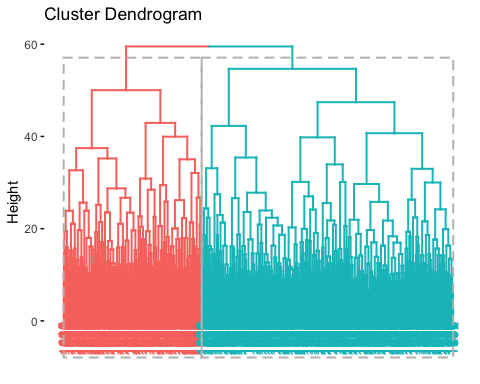
set.seed(100)  
  
# Create Dissimilarity matrix  
diss.matrix = dist(diet.data, method = "euclidean")  
  
#Identifying the optimal number of clusters given complete linkage method  
gap\_stat\_c = clusGap(diet.data, FUN = hcut, hc\_method = "complete", K.max = 10, B = 5)  
fviz\_gap\_stat(gap\_stat\_c)



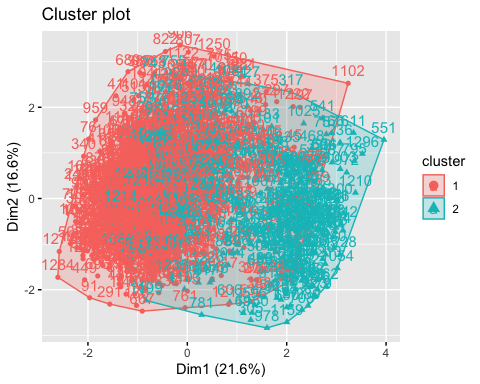
#Characterizing the clusters  
clusters.c = hcut(diet.data, k = 2, hc\_func = "hclust", hc\_method = "complete", hc\_metric = "euclidian")  
  
clusters.c$size

## [1] 840 461

fviz\_dend(clusters.c, rect = TRUE)



fviz\_cluster(clusters.c)



The optimal number of clusters within this data is *2* clusters. There are 840 observations in the first cluster, and 461 observations in the second cluster.

input.feature.vals = cbind(diet.data, cluster = clusters.c$cluster)  
input.feature.vals %>%  
 group\_by(cluster) %>%  
 summarise\_all(mean) %>%   
 knitr::kable()

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cluster | h\_cereal\_preg\_Ter | h\_dairy\_preg\_Ter | h\_fastfood\_preg\_Ter | h\_fish\_preg\_Ter | h\_fruit\_preg\_Ter | h\_legume\_preg\_Ter | h\_meat\_preg\_Ter | h\_veg\_preg\_Ter |
| 1 | 9.220534 | 26.54682 | 0.8116658 | 3.35139 | 12.91455 | 2.078733 | 8.233880 | 11.11700 |
| 2 | 29.754926 | 23.54253 | 0.6837526 | 3.07373 | 13.63977 | 1.360025 | 8.406542 | 11.16912 |

The first cluster has less cereal consumed during pregnancy by mothers than the second cluster; respondents also consumed slightly more dairy, fast food, fish and legumes. Those in the second cluster consumed more cereal and slightly more fruit and meats. Both clusters consumed approximately the same amount of vegetables.

# Part 2: Choose your own supervised adventure

## Research question

The goal of this analysis is to generate hypotheses regarding dietary patterns and toxic metals during pregnancy on child externalizing and internalizing behaviors. This hypotheses generating analysis will guide future research on the role of nutrition on child behaviors, and whether interactions between dietary patterns and toxic metals are salient for child behaviors. This is being done to generate hypotheses regarding the importance of both toxins in the built environment and maternal nutrition on child well-being, and to gain a better understanding of how these factors interact in relation to child health outcomes.

## Loading and preparing data

Loading data into single data frame.

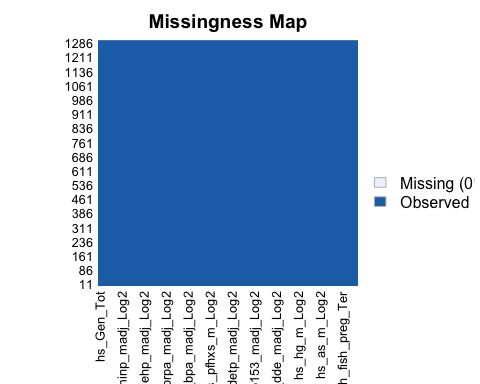
#Load data using path of where file is stored  
load("./data/exposome.RData")  
  
studydata =   
 merge(exposome,phenotype,by = "ID") %>%   
 merge(covariates, by = "ID")  
  
studydata$ID = NULL

Data cleaning and selecting relevant features to research question. The outcome of interest, the CBCL scale assessing child internalizing and externalizing behaviors, was dichotomized at a cut-off of 60 which has been shown to be a clinically relevant score (Biederman et al., 1993).

studydata = studydata %>%   
 select(h\_cereal\_preg\_Ter,  
 h\_dairy\_preg\_Ter,  
 h\_fastfood\_preg\_Ter,  
 h\_fish\_preg\_Ter,  
 h\_folic\_t1\_None,  
 h\_fruit\_preg\_Ter,  
 h\_legume\_preg\_Ter,  
 h\_meat\_preg\_Ter,  
 hs\_as\_m\_Log2,  
 hs\_cd\_m\_Log2,  
 hs\_co\_m\_Log2,  
 hs\_cs\_m\_Log2,  
 hs\_cu\_m\_Log2,  
 hs\_hg\_m\_Log2,  
 hs\_mn\_m\_Log2,  
 hs\_mo\_m\_Log2,  
 hs\_pb\_m\_Log2,  
 hs\_tl\_mdich\_None,  
 hs\_dde\_madj\_Log2,  
 hs\_ddt\_madj\_Log2,  
 hs\_hcb\_madj\_Log2,  
 hs\_pcb118\_madj\_Log2,  
 hs\_pcb138\_madj\_Log2,  
 hs\_pcb153\_madj\_Log2,  
 hs\_pcb170\_madj\_Log2,  
 hs\_pcb180\_madj\_Log2,  
 hs\_sumPCBs5\_madj\_Log2,  
 hs\_dep\_madj\_Log2,  
 hs\_detp\_madj\_Log2,  
 hs\_dmp\_madj\_Log2,  
 hs\_dmtp\_madj\_Log2,  
 hs\_pbde153\_madj\_Log2,  
 hs\_pbde47\_madj\_Log2,  
 hs\_pfhxs\_m\_Log2,  
 hs\_pfna\_m\_Log2,  
 hs\_pfoa\_m\_Log2,  
 hs\_pfos\_m\_Log2,  
 hs\_pfunda\_m\_Log2,  
 hs\_bpa\_madj\_Log2,  
 hs\_bupa\_madj\_Log2,  
 hs\_etpa\_madj\_Log2,  
 hs\_mepa\_madj\_Log2,  
 hs\_oxbe\_madj\_Log2,  
 hs\_prpa\_madj\_Log2,  
 hs\_trcs\_madj\_Log2,  
 hs\_mbzp\_madj\_Log2,  
 hs\_mecpp\_madj\_Log2,  
 hs\_mehhp\_madj\_Log2,  
 hs\_mehp\_madj\_Log2,  
 hs\_meohp\_madj\_Log2,  
 hs\_mep\_madj\_Log2,  
 hs\_mibp\_madj\_Log2,  
 hs\_mnbp\_madj\_Log2,  
 hs\_ohminp\_madj\_Log2,  
 hs\_oxominp\_madj\_Log2,  
 hs\_sumDEHP\_madj\_Log2,  
 e3\_asmokcigd\_p\_None,  
 hs\_cotinine\_mcat\_None,  
 hs\_Gen\_Tot  
 )   
  
data.rec = studydata %>%   
 mutate(  
 behav = cut(hs\_Gen\_Tot,   
 breaks = c(-Inf, 60, Inf),  
 labels = c("0", "1"))  
 ) %>%   
 select(-hs\_Gen\_Tot) %>%   
 select(everything(), behav)

Data exploration.

# Investigating missing data  
missmap(studydata)



# No missingness observed.  
  
# Exploring continuous/categorical variables of interest  
table.1 = tableby(~ behav + h\_fastfood\_preg\_Ter + h\_dairy\_preg\_Ter +  
 h\_meat\_preg\_Ter + hs\_as\_m\_Log2 + hs\_pfoa\_m\_Log2 + hs\_cotinine\_mcat\_None,  
 data = data.rec,  
 numeric.stats = c("mean","median", "range"))  
summary(table.1, text = TRUE)

|  |  |
| --- | --- |
|  | Overall (N=1301) |
| behav |  |
| - 0 | 1235 (94.9%) |
| - 1 | 66 (5.1%) |
| h\_fastfood\_preg\_Ter |  |
| - (0,0.25] | 94 (7.2%) |
| - (0.25,0.83] | 535 (41.1%) |
| - (0.83,Inf] | 672 (51.7%) |
| h\_dairy\_preg\_Ter |  |
| - (0,17.1] | 270 (20.8%) |
| - (17.1,27.1] | 380 (29.2%) |
| - (27.1,Inf] | 651 (50.0%) |
| h\_meat\_preg\_Ter |  |
| - (0,6.5] | 427 (32.8%) |
| - (6.5,10] | 387 (29.7%) |
| - (10,Inf] | 487 (37.4%) |
| hs\_as\_m\_Log2 |  |
| - Mean | -3.011 |
| - Median | -1.925 |
| - Range | -38.625 - 6.493 |
| hs\_pfoa\_m\_Log2 |  |
| - Mean | 1.048 |
| - Median | 1.201 |
| - Range | -5.476 - 4.984 |
| hs\_cotinine\_mcat\_None |  |
| - Non-smokers | 759 (58.3%) |
| - SHS smokers | 157 (12.1%) |
| - Smokers | 385 (29.6%) |

Overall, 1301 participants are included in this dataset. Of those, 66 (5.1%) children scored above 60 on the CBCL scale, suggesting that they have internalizing and externalizing behaviors. The data is unbalanced, which will be taken into account by down-sampling.

Partitioning data.

set.seed(100)  
  
#Partition data for use in demonstration  
train.indices = createDataPartition(y = data.rec$behav, p = 0.7,list = FALSE)  
training = data.rec[train.indices, ]  
testing = data.rec[-train.indices, ]

## First Algorithm: Using LASSO for Feature Selection

Hyperparameter tuning and initial accuracy testing.

set.seed(100)  
  
#Create grid to search lambda  
lambda = 10^seq(-3,3, length = 100)  
  
lasso.m1 = train(  
 behav ~.,   
 data = training,   
 method = "glmnet",   
 trControl = trainControl("cv", number = 10, sampling = "down"),   
 tuneGrid = expand.grid(alpha = 0, lambda = lambda)  
)  
  
#Print the values of alpha and lambda that gave best prediction  
lasso.m1$bestTune

## alpha lambda  
## 27 0 0.03764936

# Model coefficients  
coef(lasso.m1$finalModel, lasso.m1$bestTune$lambda)

## 67 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 14.940860403  
## h\_cereal\_preg\_Ter(9,27.3] 1.645293264  
## h\_cereal\_preg\_Ter(27.3,Inf] -0.313519257  
## h\_dairy\_preg\_Ter(17.1,27.1] 0.564628064  
## h\_dairy\_preg\_Ter(27.1,Inf] 0.843949807  
## h\_fastfood\_preg\_Ter(0.25,0.83] 0.440350952  
## h\_fastfood\_preg\_Ter(0.83,Inf] 0.159430252  
## h\_fish\_preg\_Ter(1.9,4.1] 0.140287923  
## h\_fish\_preg\_Ter(4.1,Inf] -0.009283005  
## h\_folic\_t1\_None1 0.172552295  
## h\_fruit\_preg\_Ter(0.6,18.2] 0.414253444  
## h\_fruit\_preg\_Ter(18.2,Inf] -0.016950457  
## h\_legume\_preg\_Ter(0.5,2] 0.821974495  
## h\_legume\_preg\_Ter(2,Inf] -0.208598448  
## h\_meat\_preg\_Ter(6.5,10] -1.016132191  
## h\_meat\_preg\_Ter(10,Inf] -0.313651564  
## hs\_as\_m\_Log2 0.050451740  
## hs\_cd\_m\_Log2 0.417422397  
## hs\_co\_m\_Log2 -0.147576141  
## hs\_cs\_m\_Log2 -0.500396663  
## hs\_cu\_m\_Log2 -1.432264989  
## hs\_hg\_m\_Log2 0.201282104  
## hs\_mn\_m\_Log2 0.014783960  
## hs\_mo\_m\_Log2 -0.020125609  
## hs\_pb\_m\_Log2 -0.022180464  
## hs\_tl\_mdich\_NoneUndetected -0.241384277  
## hs\_dde\_madj\_Log2 0.006378354  
## hs\_ddt\_madj\_Log2 0.045288757  
## hs\_hcb\_madj\_Log2 -0.035752498  
## hs\_pcb118\_madj\_Log2 0.383854562  
## hs\_pcb138\_madj\_Log2 -0.303396487  
## hs\_pcb153\_madj\_Log2 -0.407879454  
## hs\_pcb170\_madj\_Log2 0.176181334  
## hs\_pcb180\_madj\_Log2 -0.067347794  
## hs\_sumPCBs5\_madj\_Log2 0.171046828  
## hs\_dep\_madj\_Log2 -0.151474807  
## hs\_detp\_madj\_Log2 -0.086505640  
## hs\_dmp\_madj\_Log2 0.146054446  
## hs\_dmtp\_madj\_Log2 -0.036573156  
## hs\_pbde153\_madj\_Log2 0.229068887  
## hs\_pbde47\_madj\_Log2 0.205832775  
## hs\_pfhxs\_m\_Log2 0.169774915  
## hs\_pfna\_m\_Log2 -0.045938167  
## hs\_pfoa\_m\_Log2 -0.265584680  
## hs\_pfos\_m\_Log2 -0.089743848  
## hs\_pfunda\_m\_Log2 -0.054111709  
## hs\_bpa\_madj\_Log2 -0.168153941  
## hs\_bupa\_madj\_Log2 0.102588964  
## hs\_etpa\_madj\_Log2 -0.057456021  
## hs\_mepa\_madj\_Log2 -0.139815974  
## hs\_oxbe\_madj\_Log2 0.116383085  
## hs\_prpa\_madj\_Log2 0.044902914  
## hs\_trcs\_madj\_Log2 0.079458183  
## hs\_mbzp\_madj\_Log2 0.073528857  
## hs\_mecpp\_madj\_Log2 -0.189141777  
## hs\_mehhp\_madj\_Log2 -0.039037253  
## hs\_mehp\_madj\_Log2 0.210975187  
## hs\_meohp\_madj\_Log2 -0.214800494  
## hs\_mep\_madj\_Log2 0.035458352  
## hs\_mibp\_madj\_Log2 -0.065340151  
## hs\_mnbp\_madj\_Log2 0.385473372  
## hs\_ohminp\_madj\_Log2 0.021850333  
## hs\_oxominp\_madj\_Log2 0.139637920  
## hs\_sumDEHP\_madj\_Log2 0.227121552  
## e3\_asmokcigd\_p\_None -0.100604164  
## hs\_cotinine\_mcat\_NoneSHS smokers -0.440304766  
## hs\_cotinine\_mcat\_NoneSmokers 0.533883438

varImp(lasso.m1)

## glmnet variable importance  
##   
## only 20 most important variables shown (out of 66)  
##   
## Overall  
## h\_cereal\_preg\_Ter(9,27.3] 100.00  
## hs\_cu\_m\_Log2 87.00  
## h\_meat\_preg\_Ter(6.5,10] 61.61  
## h\_dairy\_preg\_Ter(27.1,Inf] 51.11  
## h\_legume\_preg\_Ter(0.5,2] 49.76  
## h\_dairy\_preg\_Ter(17.1,27.1] 34.06  
## hs\_cotinine\_mcat\_NoneSmokers 32.19  
## hs\_cs\_m\_Log2 30.14  
## h\_fastfood\_preg\_Ter(0.25,0.83] 26.48  
## hs\_cotinine\_mcat\_NoneSHS smokers 26.48  
## hs\_cd\_m\_Log2 25.08  
## h\_fruit\_preg\_Ter(0.6,18.2] 24.89  
## hs\_pcb153\_madj\_Log2 24.50  
## hs\_mnbp\_madj\_Log2 23.13  
## hs\_pcb118\_madj\_Log2 23.03  
## h\_meat\_preg\_Ter(10,Inf] 18.75  
## h\_cereal\_preg\_Ter(27.3,Inf] 18.74  
## hs\_pcb138\_madj\_Log2 18.12  
## hs\_pfoa\_m\_Log2 15.82  
## hs\_tl\_mdich\_NoneUndetected 14.34

# Make predictions  
pred.lasso.1 = predict(lasso.m1, training)  
pred.lasso.prob.1 = predict(lasso.m1, training, type = "prob")  
  
# Model prediction performance  
eval.results = confusionMatrix(pred.lasso.1, training$behav, positive = "1")  
print(eval.results)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 484 4  
## 1 381 43  
##   
## Accuracy : 0.5779   
## 95% CI : (0.545, 0.6102)  
## No Information Rate : 0.9485   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.099   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.91489   
## Specificity : 0.55954   
## Pos Pred Value : 0.10142   
## Neg Pred Value : 0.99180   
## Prevalence : 0.05154   
## Detection Rate : 0.04715   
## Detection Prevalence : 0.46491   
## Balanced Accuracy : 0.73722   
##   
## 'Positive' Class : 1   
##

#Accuracy of this model is 0.58

## Second Algorithm: Elastic Net

set.seed(100)  
  
alpha = seq(0,1, length = 100)  
  
en.m = train(  
 behav ~.,   
 data = training,   
 method = "glmnet",   
 trControl = trainControl("cv", number = 10, sampling = "down"),   
 tuneGrid = expand.grid(alpha = alpha, lambda = lambda)  
)  
  
#Print the values of alpha and lambda that gave best prediction  
en.m$bestTune

## alpha lambda  
## 7802 0.7878788 0.001149757

# Model coefficients  
coef(en.m$finalModel, en.m$bestTune$lambda)

## 67 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) -4.1058594548  
## h\_cereal\_preg\_Ter(9,27.3] -2.5918971660  
## h\_cereal\_preg\_Ter(27.3,Inf] .   
## h\_dairy\_preg\_Ter(17.1,27.1] .   
## h\_dairy\_preg\_Ter(27.1,Inf] 2.6212171111  
## h\_fastfood\_preg\_Ter(0.25,0.83] -0.6330526288  
## h\_fastfood\_preg\_Ter(0.83,Inf] 2.3009155316  
## h\_fish\_preg\_Ter(1.9,4.1] 0.0005582068  
## h\_fish\_preg\_Ter(4.1,Inf] -8.1459085620  
## h\_folic\_t1\_None1 -0.5185944712  
## h\_fruit\_preg\_Ter(0.6,18.2] 2.2375829722  
## h\_fruit\_preg\_Ter(18.2,Inf] -2.0413379172  
## h\_legume\_preg\_Ter(0.5,2] 0.8301769393  
## h\_legume\_preg\_Ter(2,Inf] 10.7929584527  
## h\_meat\_preg\_Ter(6.5,10] 0.0018075878  
## h\_meat\_preg\_Ter(10,Inf] 1.5135998487  
## hs\_as\_m\_Log2 .   
## hs\_cd\_m\_Log2 .   
## hs\_co\_m\_Log2 -1.4214239696  
## hs\_cs\_m\_Log2 -3.9916843193  
## hs\_cu\_m\_Log2 -0.3202327644  
## hs\_hg\_m\_Log2 2.6313548564  
## hs\_mn\_m\_Log2 -2.2404134428  
## hs\_mo\_m\_Log2 0.3488430253  
## hs\_pb\_m\_Log2 -1.0312966456  
## hs\_tl\_mdich\_NoneUndetected .   
## hs\_dde\_madj\_Log2 -0.4158867097  
## hs\_ddt\_madj\_Log2 0.3937276284  
## hs\_hcb\_madj\_Log2 0.7696578095  
## hs\_pcb118\_madj\_Log2 2.2646359536  
## hs\_pcb138\_madj\_Log2 -2.4467355052  
## hs\_pcb153\_madj\_Log2 .   
## hs\_pcb170\_madj\_Log2 .   
## hs\_pcb180\_madj\_Log2 .   
## hs\_sumPCBs5\_madj\_Log2 .   
## hs\_dep\_madj\_Log2 .   
## hs\_detp\_madj\_Log2 .   
## hs\_dmp\_madj\_Log2 0.4933327478  
## hs\_dmtp\_madj\_Log2 -0.6627828823  
## hs\_pbde153\_madj\_Log2 1.3355889084  
## hs\_pbde47\_madj\_Log2 0.1464316124  
## hs\_pfhxs\_m\_Log2 -0.5353491580  
## hs\_pfna\_m\_Log2 1.0667231638  
## hs\_pfoa\_m\_Log2 0.8850638179  
## hs\_pfos\_m\_Log2 0.3779620390  
## hs\_pfunda\_m\_Log2 -2.1954025206  
## hs\_bpa\_madj\_Log2 -1.3711761419  
## hs\_bupa\_madj\_Log2 -0.2592824523  
## hs\_etpa\_madj\_Log2 0.0764868758  
## hs\_mepa\_madj\_Log2 0.5020566858  
## hs\_oxbe\_madj\_Log2 -0.1834275537  
## hs\_prpa\_madj\_Log2 .   
## hs\_trcs\_madj\_Log2 -0.2353718024  
## hs\_mbzp\_madj\_Log2 -1.1964755320  
## hs\_mecpp\_madj\_Log2 1.3034335995  
## hs\_mehhp\_madj\_Log2 -0.1136900396  
## hs\_mehp\_madj\_Log2 0.5239053317  
## hs\_meohp\_madj\_Log2 1.9674342670  
## hs\_mep\_madj\_Log2 0.1846658509  
## hs\_mibp\_madj\_Log2 -0.0796975949  
## hs\_mnbp\_madj\_Log2 -0.6450291373  
## hs\_ohminp\_madj\_Log2 -1.3662112473  
## hs\_oxominp\_madj\_Log2 0.7122665335  
## hs\_sumDEHP\_madj\_Log2 .   
## e3\_asmokcigd\_p\_None -0.0003709927  
## hs\_cotinine\_mcat\_NoneSHS smokers 2.0383235491  
## hs\_cotinine\_mcat\_NoneSmokers 0.8542630058

varImp(en.m)

## glmnet variable importance  
##   
## only 20 most important variables shown (out of 66)  
##   
## Overall  
## h\_legume\_preg\_Ter(2,Inf] 100.00  
## h\_fish\_preg\_Ter(4.1,Inf] 75.47  
## hs\_cs\_m\_Log2 36.98  
## hs\_hg\_m\_Log2 24.38  
## h\_dairy\_preg\_Ter(27.1,Inf] 24.29  
## h\_cereal\_preg\_Ter(9,27.3] 24.01  
## hs\_pcb138\_madj\_Log2 22.67  
## h\_fastfood\_preg\_Ter(0.83,Inf] 21.32  
## hs\_pcb118\_madj\_Log2 20.98  
## hs\_mn\_m\_Log2 20.76  
## h\_fruit\_preg\_Ter(0.6,18.2] 20.73  
## hs\_pfunda\_m\_Log2 20.34  
## h\_fruit\_preg\_Ter(18.2,Inf] 18.91  
## hs\_cotinine\_mcat\_NoneSHS smokers 18.89  
## hs\_meohp\_madj\_Log2 18.23  
## h\_meat\_preg\_Ter(10,Inf] 14.02  
## hs\_co\_m\_Log2 13.17  
## hs\_bpa\_madj\_Log2 12.70  
## hs\_ohminp\_madj\_Log2 12.66  
## hs\_pbde153\_madj\_Log2 12.37

# Make predictions  
pred.en = predict(en.m, training)  
pred.en.prob = predict(en.m, training, type = "prob")  
  
# Model prediction performance  
eval.results = confusionMatrix(pred.en, training$behav, positive = "1")  
print(eval.results)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 401 0  
## 1 464 47  
##   
## Accuracy : 0.4912   
## 95% CI : (0.4583, 0.5242)  
## No Information Rate : 0.9485   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.0818   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 1.00000   
## Specificity : 0.46358   
## Pos Pred Value : 0.09198   
## Neg Pred Value : 1.00000   
## Prevalence : 0.05154   
## Detection Rate : 0.05154   
## Detection Prevalence : 0.56031   
## Balanced Accuracy : 0.73179   
##   
## 'Positive' Class : 1   
##

#Accuracy of this model is 0.49

Given that the LASSO model produced slightly better accuracy than the elastic net model, the LASSO model will be used for final accuracy testing.

### Final accuracy testing

set.seed(100)  
  
# Using best fit model from above with testing data  
pred.lasso.f = predict(lasso.m1, testing)  
pred.lasso.f.prob = predict(lasso.m1, testing, type = "prob")  
  
# Evaluating in testing data with confusion matrix  
eval.results = confusionMatrix(pred.lasso.f, testing$behav, positive = "1")  
print(eval.results)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 181 7  
## 1 189 12  
##   
## Accuracy : 0.4961   
## 95% CI : (0.4454, 0.547)  
## No Information Rate : 0.9512   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.0218   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.63158   
## Specificity : 0.48919   
## Pos Pred Value : 0.05970   
## Neg Pred Value : 0.96277   
## Prevalence : 0.04884   
## Detection Rate : 0.03085   
## Detection Prevalence : 0.51671   
## Balanced Accuracy : 0.56038   
##   
## 'Positive' Class : 1   
##

Final accuracy testing shows that the accuracy of the LASSO model is 0.50. The LASSO model suggests that the most important variable is medium cereal consumption, followed by copper blood levels, medium meat consumption, high dairy consumption, and medium legume consumption. The elastic net model suggests that the most important variable is high legume consumption, high fish consumption, cesium blood levels, mercury blood levels, and high dairy consumption. Given that the LASSO model had greater accuracy, emphasis will be placed on those variables for future exploration. However, given that legume and dairy consumption were both among the top variables of importance in both the elastic net and LASSO models, there is support to explore these dietary components in greater detail.

# Part 3: Ethical considerations of data-driven analysis in social epidemiology

## Question 1

*Braithwaite SR, Giraud-Carrier C, West J, Barnes MD, Hanson CL. Validating machine learning algorithms for Twitter data against established measures of suicidality. JMIR Ment Health, 2016: 3(2): e21.*

#### Quality of analysis

The paper that I’ve selected to analyze is the Braithwaite et al. (2016) paper predicting suicide from Twitter data. Overall, I do not think this paper is helpful in predicting early detection of suicidality. One thing that immediately stood out to me pertaining to the quality of analysis is the data quality. We’ve talked a lot in class about “garbage in, garbage out” and I think this paper is a good example of issues that arise when you use problematic data. Firstly, the authors explain that they carried out study recruitment at two different time periods, as initial recruitment did not result in a sufficient number of participants. This is important given that their exclusion criteria involved needing to tweet within the last month, and participants recruited in the initial stage could theoretically not have been included in the second phase. Unfortunately, the authors do not have the power to stratify by recruitment period or take into account any time component.

The authors also opted to use a decision tree algorithm for feature selection and variable importance. It would have been interesting to see how this compares to LASSO. The authors found that tweeting in the “achieve”, “religion” or “relativity” categories are predictive for being nonsuicidal. These seem very arbitrary, particularly since the “relativity” category is related to theories of relativity, and do not seem to be trustworthy features of importance when considering suicidality.

#### Bias

The authors describe recruiting through Mechanical Turk (MTurk), which is an Amazon-led form of task deployment where workers can sign up to do tasks for a negligible amount of money. As described in the paper, participants were paid the “market rate”, which was a shocking 30-50 cents. Immediately, this should be cause for concern; people willing to participate in a study for a mere 30 cents are likely not representative of the general population or the average social media user in the US. Therefore, while the authors frame this study as being relevant for the US population, I don’t believe that is accurate.

To understand potential bias, the authors should have presented more detailed sociodemographic characteristics. The authors fail to report the average age of the participants, which would be important when understanding suicide, particularly as age is a risk factor. Additionally, the authors don’t present information on sex. As men are at higher risk of suicide than women, this information would have been important to present as well for understanding potential bias.

#### Details of methodology

At the recruitment stage, the authors excluded 77 participants without explanation for why these participants were excluded. They do present some exclusion criteria, but it would have been more transparent to explain clearly why participants were excluded.

The authors failed to describe how they split their data and whether they split 70/30. They also failed to describe whether and how they varied their hyperparameters, which would have been “cp” in this algorithm. They do mention that they used leave-one-out cross-validation, but don’t mention what they set their cross-validation parameter as, and instead note that they repeated cross-validation N times. Lastly, they should have taken into account the unbalanced data but do not describe any approach in doing so.

## Question 2

In this paper, the authors suggest creating some type of system that includes text messaging to reach individuals who the algorithm identify as being at risk of suicide. However, given that the algorithm identified somewhat arbitrary text and was trained on a small number of users, it seems that it may misidentify users at risk of suicide. If this algorithm is implemented in real time, it could be harmful if users are flagged for suicidal behavior without actually exhibiting suicidal behavior. This could lead to self-fulfilling behavior, in that being told they are at risk of suicide, or repeatedly reminded of suicide, actually places them at higher risk of suicide.

At an ecological level, there exist disparities in internet access across the United States. Counties or neighborhoods could be identified as having users with high rates of suicide, or high rates of suicidal behavior, and resources could be funneled towards these neighborhoods. Counties with lower internet access and fewer users, such as those in rural areas, may not be allocated resources because they were not identified as having users with high suicide rates. Given associations between rurality and suicide, this could be potentially harmful for rural communities who are in need of these resources.

## Question 3

Safeguards at the individual level would involve ensuring that users could opt-out of any possible intervention. By doing so, users would not be subject to unwanted text messages about potential suicidal behavior and avoid triggering social media users and putting them in an emotionally vulnerable state.

At the policy level, safeguards should include not making funding decisions based solely from data collected via social media. Policymakers should be well-versed in understanding disparities in internet access, and should not consider allocating funding using results from analyses of social media.

# References

Biderman J, Faraone SV, Doyle A, Lehman BK, Kraus I, Perrin J, Tsuang MT. Convergence of the Child Behavior Checklist with structured interview-based psychiatric diagnoses of ADHD children with and without comorbidity. J Child Psychiat, 1993: 34(7); 1241-1251.

Braithwaite SR, Giraud-Carrier C, West J, Barnes MD, Hanson CL. Validating machine learning algorithms for Twitter data against established measures of suicidality. JMIR Ment Health, 2016: 3(2): e21.