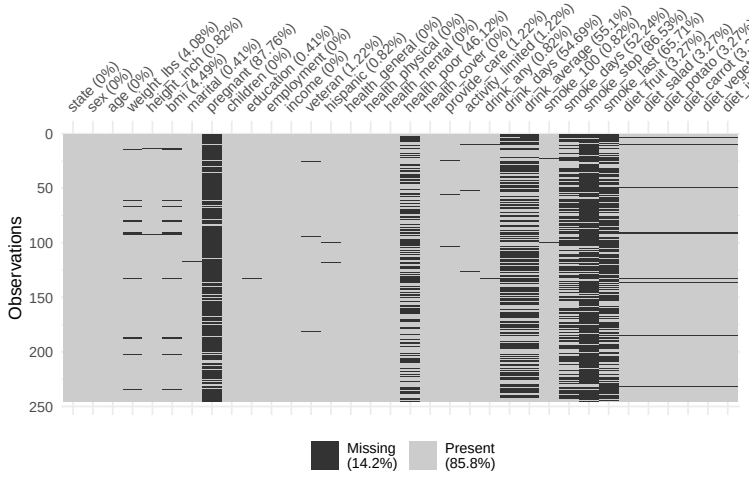
M445 HW 8

Erick Castillo

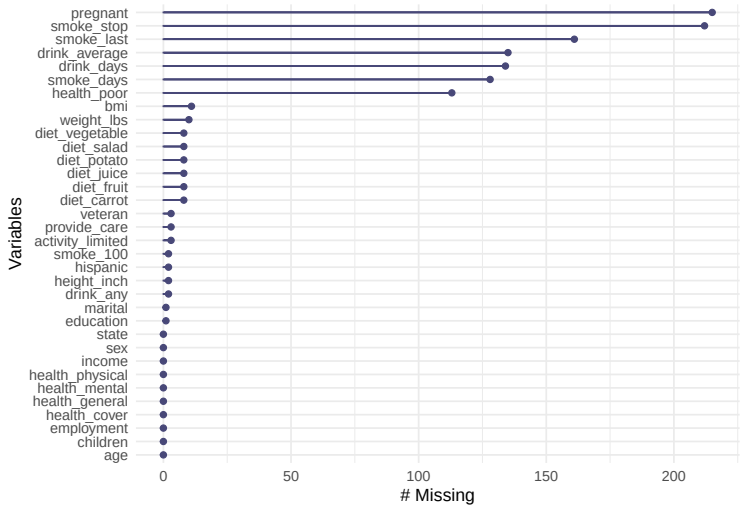
4/16/2021

Part A: Use data visualization techniques to explore the patterns of the missing data. Write down some conclusions.

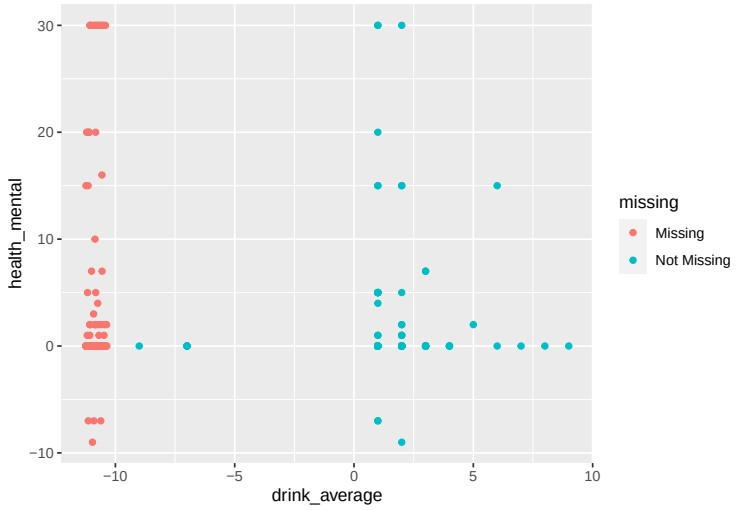
```
vis_miss(df)
```




```
gg_miss_var(df)
```



```
ggplot(df,
       aes(x = drink_average,
           y = health_mental)) +
  geom_miss_point()
```



The above plots say a lot about the missingness of the data. The first one indicates that 14.2% of the total data is missing. The second plot indicates that the top 7 columns in the data frame make up the majority of the missing data. I selected one of these arbitrarily to then created the third and final plot. This indicates that with relation to the response variable, the predictor has most of the missing values. The majority of the values present in the data frame have values greater than 0.

Part B: Does the data appear to be MAR?

Yes. There does not seem to be an apparent pattern to the missing data. This means that a variety of imputation methods may be used.

Part C. Impute the missing data for the bmi column using the mean of the columns, the regression of bmi on all the diet related predictors, stochastic regression on all the diet predictors.

```
#inputting the mean for NAs.
mu.var <- df$bmi
mu.var[is.na(mu.var)] <- mean(mu.var, na.rm = TRUE)
mu.var.stats = c(mean(mu.var), sd(mu.var), cor(mu.var, df$diet_fruit, use = 'complete.obs'))

#regression of bmi on all diet relate predictors
id_na = which(is.na(df$bmi))
data_na = df[id_na,]
obs_data = df[-id_na,]

newdata = data.frame(data_na[,6], data_na[,29:34])

mod1 = lm(bmi~diet_fruit+diet_salad+diet_potato+diet_carrot+diet_vegetable+diet_juice, data = obs_data)
predict1 = predict(mod1, newdata = newdata)

reg.var <- df$bmi
```

```
reg.var[is.na(reg.var)] <- predict1 #two cells still NA.
reg.var.stats = c(mean(reg.var, na.rm = TRUE), sd(reg.var, na.rm = TRUE), cor(reg.var, df$diet_fruit, us

#stochastic regression.
cols.stoch = data.frame(df[,'bmi'], df[,29:34])
imp.stoch = mice(cols.stoch, method = 'norm.nob', m=1, maxit=1, seed=1,print=FALSE)
imp.cols = complete(imp.stoch)
mice.var.stats = c(mean(imp.cols$df....bmi..), sd(imp.cols$df....bmi..), cor(imp.cols$df....bmi.., imp.c

#dataframe comparing the statistics side-by-side
data.frame(mu.var.stats, reg.var.stats, mice.var.stats)
```

```
##   mu.var.stats reg.var.stats mice.var.stats
## 1   27.7845726    27.7612124     27.7721910
## 2    6.4192448     6.4611370      6.4845188
## 3   -0.1092105    -0.1196657     -0.1063432
```

The outputted dataframe displays the mean in the first row, the standard deviation in the second, and correlation in the third. The first column pertains to simply imputing the mean, the second to regression, and the third to stochastic regression. Notice that all the rows tend to stick around the same values.

Part D. Use MICE to impute the BMI column using the entire data set and $m = 20$ multiple imputations.

```
#creation of matrix.
df.bool <- is.na(df)
df.bool[,-6] <- FALSE

imp2 = mice(df, m=20, methods = 'pmm', maxit = 20, where = df.bool, print = FALSE)
wholedf = complete(imp2,10)
micewhole.var.stats = c(mean(wholedf$bmi, na.rm = TRUE), sd(wholedf$bmi, na.rm=TRUE), cor(wholedf$bmi, w

data.frame(mu.var.stats, reg.var.stats, mice.var.stats, micewhole.var.stats)
```

```
##   mu.var.stats reg.var.stats mice.var.stats micewhole.var.stats
## 1   27.7845726    27.7612124     27.7721910          27.7845726
## 2    6.4192448     6.4611370      6.4845188           6.5690247
## 3   -0.1092105    -0.1196657     -0.1063432          -0.1113397
```

Notice that the mean in this case is the same as when simply imputing the mean; however, the increase in the standard deviation indicates that the values have more variety than in all the other cases. This would be my preferred method of imputation.

Part E. Now impute all the missing values using MICE, and use this dataset to run a model of my choice.

```
imp3 = mice(df, m=1, maxit=1, seed=1, method = c('','','','pmm','pmm','pmm','polr','logreg',
                          '','polr','','','polr','logreg','','','',
                          'pmm','','logreg','logreg','logreg','pmm','pmm','logreg', 'polr',
                          'logreg','polr','pmm','pmm','pmm','pmm','pmm','pmm'))
```

```
##
##  iter imp variable
##   1   1  weight_lbs  height_inch  bmi  marital  pregnant  education  veteran  hispanic

## Warning: glm.fit: algorithm did not converge

##   health_poor  provide_care  activity_limited  drink_any  drink_days*  drink_average*  smoke_100  sm

## Warning: Number of logged events: 17
```

```
mice_df = complete(imp3)
mice_df$state <- as.integer(mice_df$state)
mod2 = lm(health_mental~., data = mice_df)
summary(mod2)
```

```
##
## Call:
## lm(formula = health_mental ~ ., data = mice_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.4880  -3.4227  -0.8592   1.9299  22.5259
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -63.690685  45.839870  -1.389  0.16644
## state                     0.020686   0.038599   0.536  0.59269
## sexFemale                 3.808905   1.882686   2.023  0.04456 *
## age                      -0.016804   0.046978  -0.358  0.72099
## weight_lbs               -0.078824   0.110100  -0.716  0.47497
## height_inch               0.739803   0.649855   1.138  0.25648
## bmi                       0.441694   0.688567   0.641  0.52204
## maritalDivorced           2.339285   1.705812   1.371  0.17199
## maritalWidowed           -1.586161   1.794578  -0.884  0.37796
## maritalSeparated          8.636416   5.609074   1.540  0.12540
## maritalNeverMarried       3.249368   2.121811   1.531  0.12744
## maritalUnmarriedCouple    0.042558   3.798474   0.011  0.99107
## pregnantNo                0.621686   1.456749   0.427  0.67007
## children                  1.515610   0.785577   1.929  0.05528 .
## education2               10.713325   9.809265   1.092  0.27624
## education3               10.736752   9.747578   1.101  0.27218
## education4                9.370254   9.723822   0.964  0.33653
## education5                9.522863   9.680553   0.984  0.32659
## education6                6.122631   9.836670   0.622  0.53446
## employment2              -1.366604   2.074438  -0.659  0.51089
## employment3               6.660204   3.912690   1.702  0.09046 .
## employment4               2.360195   2.895966   0.815  0.41616
## employment5              -3.753778   2.506959  -1.497  0.13608
## employment7              -0.612506   1.643500  -0.373  0.70983
## employment8              -1.576874   2.595865  -0.607  0.54432
## income10-15k              0.459779   3.418828   0.134  0.89317
## income15-20k             -4.273464   3.626712  -1.178  0.24024
## income20-25k             -1.722982   3.491215  -0.494  0.62225
```

```
## income25-35k            -2.114728   3.312927  -0.638  0.52408
## income35-50k            -0.591221   3.201591  -0.185  0.85370
## income50-75k            -0.535879   3.434052  -0.156  0.87617
## income>75k               0.071767   3.409055   0.021  0.98323
## incomeDontknow          -0.525610   3.511189  -0.150  0.88117
## incomeRefused           -0.923031   3.471175  -0.266  0.79061
## veteran2                 5.052936   9.577396   0.528  0.59844
## veteran3                 6.634687   8.642585   0.768  0.44370
## veteran4                 6.563596   9.122230   0.720  0.47277
## veteran5                 4.328796   8.384101   0.516  0.60628
## hispanicNo               2.154211   2.228252   0.967  0.33497
## health_generalVeryGood  -0.570035   1.729749  -0.330  0.74213
## health_generalGood       0.464977   1.751036   0.266  0.79090
## health_generalFair       3.691586   2.441392   1.512  0.13229
## health_generalPoor       9.727798   3.288845   2.958  0.00352 **
## health_generalRefused   -7.871254   6.437445  -1.223  0.22305
## health_physical          0.128103   0.083060   1.542  0.12478
## health_poor             -0.002318   0.058330  -0.040  0.96834
## health_coverNo          -1.134436   1.810306  -0.627  0.53169
## provide_careNo          -0.534115   1.244198  -0.429  0.66824
## activity_limitedNo       1.110780   1.544450   0.719  0.47296
## drink_anyNo              1.213044   1.564044   0.776  0.43903
## drink_days               0.061244   0.059884   1.023  0.30783
## drink_average           -0.022631   0.140311  -0.161  0.87205
## smoke_100No             -1.417638   1.452032  -0.976  0.33023
## smoke_daysSomedays      -2.542219   1.806001  -1.408  0.16098
## smoke_daysNot@All       -3.508781   1.620202  -2.166  0.03167 *
## smoke_stopNo             0.558289   1.411561   0.396  0.69294
## smoke_last4             -0.171136   2.753805  -0.062  0.95052
## smoke_last5             -0.167821   2.499708  -0.067  0.94655
## smoke_last6              1.530188   2.521578   0.607  0.54473
## smoke_last7              0.694357   1.744574   0.398  0.69110
## smoke_last8              1.322035   1.990431   0.664  0.50742
## diet_fruit               0.001739   0.001962   0.886  0.37675
## diet_salad               0.005091   0.004927   1.033  0.30283
## diet_potato             -0.008419   0.004674  -1.801  0.07336 .
## diet_carrot             -0.003193   0.005875  -0.544  0.58747
## diet_vegetable           0.002765   0.001993   1.387  0.16718
## diet_juice              -0.001515   0.002013  -0.753  0.45267
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.373 on 178 degrees of freedom
## Multiple R-squared:  0.4086, Adjusted R-squared:  0.1894
## F-statistic: 1.864 on 66 and 178 DF,  p-value: 0.0006685
```

Notice that the above summary table is for the prediction of the response variable corresponding to mental health. There are a lot of categories in the dataframe. That's why the summary table is so long. the $R^2_{Adj} = 0.1894$ which is very small. This means that a little less than 20% of the variability in the dataset is explained with this least squares model.