Exploring Changes Throughout the 21st Century Between GNE, Malnourished Population Percentage, and Unemployment Rate

Jordan Toler, Tim Giang, and Elizabeth Lam

2022-12-06

Introduction

We initially wanted to obtain datasets to explore the relationships between food, food insecurity, and prices. However, due to not meeting the initial dataset constraints, we pivoted and decided to develop datasets instead, using data housed in the World Development Indicators database (*World Development Indicators.org)*. These variables were selected because the potential relationships amongst the datasets could be interesting and valuable in shaping how we examine the effect that standard country metrics could be affected by or related to how much of the country's population is malnourished.

To tidy the datasets, we had to condense the individually broken-up year columns into a singular one using pivoting functions. After this, string replacement was needed to format the year columns to not include extraneous information in the entries and to have them simply in the year digit format. From there, we re-coded column entries of numeric columns to be numeric rather than characters. We then removed redundant columns that did not contribute new information to the overall dataset. Lastly, we removed entries of the format ".." as this was how the database decided to encode missing values.

Some potential relationships we expect to see in our datasets are that a country will tend to have a lower life expectancy and a higher malnourishment percentage if its Gross National Expenditure (GNE) is lower. A country with a high percentage of unemployment will be positively correlated with a lower GNE. Our paper will explore the following research questions: Firstly, can a country's average lifespan be classified as above or below the average lifespan in 2019 using its percentage of the malnourished population and unemployment rate? Secondly, can a country's average lifespan be classified as above or below the average using all a country's GNE, unemployment rate, malnourished population percentage, year ?

```
#Load Packages
library(tidyverse)
library(readr)
library(ggplot2)
library(factoextra)
library(cluster)
library(readr)
library(plotROC)

#load datasets
gne <- read_csv("gdp.csv", show_col_types = FALSE)
life_expec_malnourish <- read_csv("life_expec_malnourish.csv", show_col_types = FALSE)
perc_unemplyed <- read_csv("perc_unemplyed.csv", show_col_types = FALSE)
```

Tidying the Datasets

The tidying process was relatively uncomplicated. First, we had to condense the individually broken-up year columns into a singular one using pivoting functions. After this, string replacement was needed to format the year columns to not include extraneous information in the entries and to have them simply in the year digit format. From there, we re-coded column entries of numeric columns to be numeric rather than characters. We then removed redundant columns that did not contribute to the overall dataset. Lastly, we removed entries of the format ".." as this was how the dataset decided to encode missing values.

```
# GDP dataset tidying --> Result: Tidy dataset w/o NA's --------------------------------
gne_clean_tidy <- gne %>%
   # remove the series code and series name column -> redundant information
   select(-`Series Code`) %>%
   select(-`Series Name`) %>%

   # combine multiple year cols. into one
   # the range of years spans from 2001 to 2020
   pivot_longer(col = c("2001 [YR2001]":"2020 [YR2020]"),
                names_to = "year",
                values_to = "gne") %>%

   # removes rows with a ".." entry on it = how NA's were coded in the sedate
   filter(!(gne == "..")) %>%

   # format the year entries: ex: want 2000 not 2000 [YR2000]
   mutate(year = str_replace_all(year,".\\[.......", "")) %>%

   # recode GNE and year entries as numeric
   mutate(year = as.numeric(year), gne = as.numeric(gne)) %>%

   # recode col titles with confusing names to have clearer titles w/o spaces
   rename(country = `Country Name` , country_code = `Country Code`)


# Life Expectancy & Nourishment dataset tidying --> Result: Tide dataset done w/o NA's -
life_clean_tidy <- life_expec_malnourish %>%

   # remove series code due to containing redundant information
   select(-`Series Code`) %>%

   # recode series name to instead be type as in type of information
   rename(type = `Series Name`) %>%

   # do year tidying to make all year cols. info contained in one
   pivot_longer(col = c("2001 [YR2001]":"2019 [YR2019]"), #
                names_to = "year",
               values_to = "stats") %>%

   # removes rows with a ".." entry on it = how NAs were coded in the datset
   filter(!(stats == '..')) %>%

   # used str replacement properly format the year entries as
   # we did in the gne dataset above
   mutate(year = str_replace_all(year,".\\[.......", "")) %>%
```

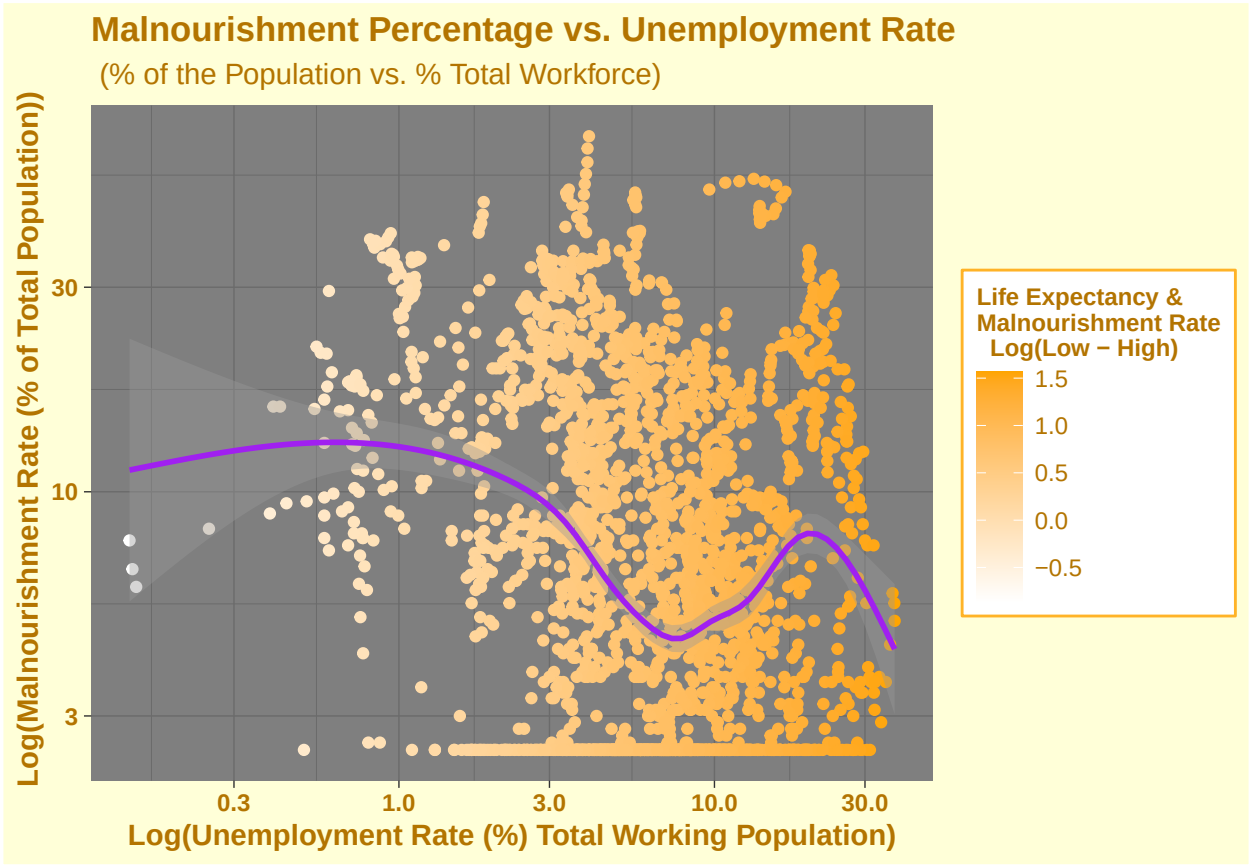
```
   # recode to make year and stats entries all numeric
   mutate(year= as.numeric(year), stats = as.numeric(stats)) %>%

   # recode col. titles to be more succinct and clear
   mutate(type = recode(type,"Life expectancy at birth, total (years)" =                                    "li
            "Prevalence of undernourishment (% of population)" = "malnourish_perc")) %>%

   # make cols. filled with corresponding data for a
   # country's life expectancy in years for a particular
   # year and a country's % of the population that is
   # malnourished for a particular year
   # The possible year range is from 2001 - 2019
   pivot_wider(names_from = type,
               values_from = stats) %>%

   # rename country and country code cols. into a
   # lowercase and w/o space format for ease of usage
   rename(country = `Country Name` , country_code = `Country Code`)


# % Unemployed dataset tidying --> Result: Tide dataset done w/o NA's ------------------
unemplyed_clean_tidy <- perc_unemplyed %>%

   # remove the series code and series name column
   # due to containing redundant information
   select(-`Series Code`) %>%
   select(-`Series Name`) %>%

   # combine every individual year column into a singular one:
   # the year range is 2001 - 2019
   pivot_longer(col = c("2001 [YR2001]":"2019 [YR2019]"),
                names_to = "year",
               values_to = "unemploy_rate") %>%

   # removes rows with a ".." entry on it = how NAs were coded in the datset
   filter(!(unemploy_rate == '..')) %>%

   # used str replacement properly format the year entries as we
   # did in the gdp dataset above
   mutate(year = str_replace_all(year,".\\[.......", "")) %>%

   # encode the entries in year and unemploy_rate to be numeric
   mutate(year= as.numeric(year), unemploy_rate = as.numeric(unemploy_rate)) %>%


   # rename country and country code cols. into a lowercase
   # and w/o space format for ease of usage
   rename(country = `Country Name` , country_code = `Country Code`)


#----- Joining ---------------------------------------------------------------------

   # first view what is missing between the datasets
```

```
  missing <- anti_join(gne_clean_tidy,unemplyed_clean_tidy, by = "country")

  # stored the dataframe of missing information to a variable:
  # this will be valuable for subsetting
  missing_countries <- missing$country


  # remove the countries in the gne dataset that
  # were not found in the missing_countrries dataframe
  gne_clean_tidy <- gne_clean_tidy %>%
  filter(!(country %in% missing_countries))


  # left join the gne dataset to the unemployment data set using three variables:
  # country, country code and year!
  final_dataset <- left_join(gne_clean_tidy,unemplyed_clean_tidy,
                             by = c("country","country_code"  ,"year"))


  # left join the life expectancy + percent of the pop. malnourished dataset
  # to the one we just created on line 142
  final_dataset <- left_join(final_dataset,life_clean_tidy, by = c("country",
                                                                   "country_code",
                                                                   "year"))

  # remove NAs that remained after completely joining all three datasets
  final_dataset <- final_dataset %>%
    drop_na()
```

Exploratory Data Analysis

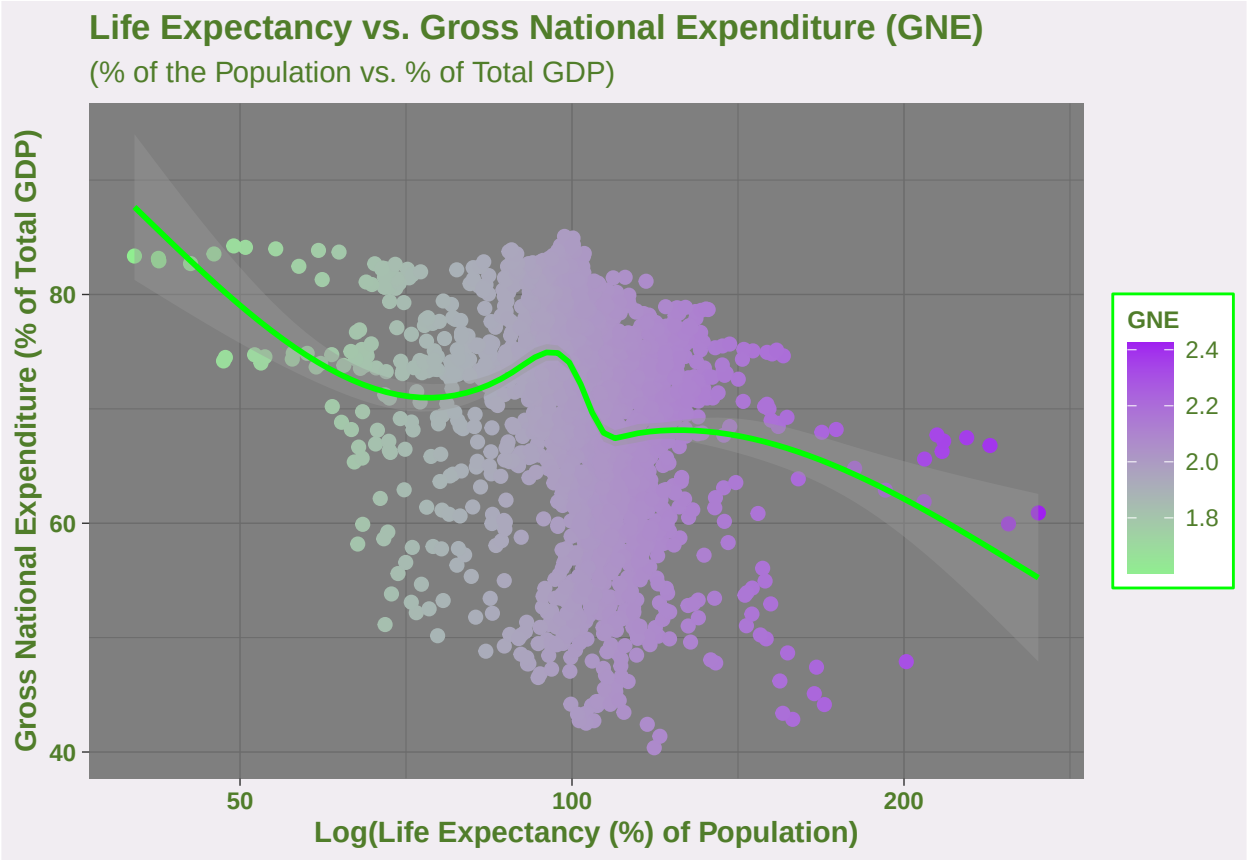
```
# Vis. 1-------------------------------------------------------------------------------
final_dataset %>%
  # map unemployment rate to malnourishment percentage
  ggplot(aes(unemploy_rate,malnourish_perc, color = ..x..)) +
  scale_x_log10() + # scale x-axis using log transform
  scale_y_log10() + # scale y-axis using log transform
  geom_point() +
  geom_smooth(colour = "purple") + # color geom_smooth line
  # add labs
  labs(title = "Malnourishment Percentage vs. Unemployment Rate",
       subtitle = " (% of the Population vs. % Total Workforce)",
       x = "Log(Unemployment Rate (%) Total Working Population)",
       y = " Log(Malnourishment Rate (% of Total Population))",
       color = "Life Expectancy &
Malnourishment Rate
  Log(Low - High)") +
  # add gradient color and stylings
    scale_color_gradient(low = "white", high = "orange") +
    theme_dark() +
    theme(plot.background=element_rect(fill="#ffffd8"),
```

```
        legend.background = element_rect(colour="#ffb327",fill="white",
                                         size=.5, linetype="solid"),
        axis.text = element_text(colour = "#b37400", face= "bold"),
        axis.title = element_text(colour = "#b37400", face= "bold"),
        legend.text = element_text(colour = "#b37400"),
        legend.title = element_text(colour = "#b37400", face = "bold", size = 9),
        plot.title = element_text(colour = "#b37400", face = "bold"),
        plot.subtitle = element_text(colour = "#b37400"))
```

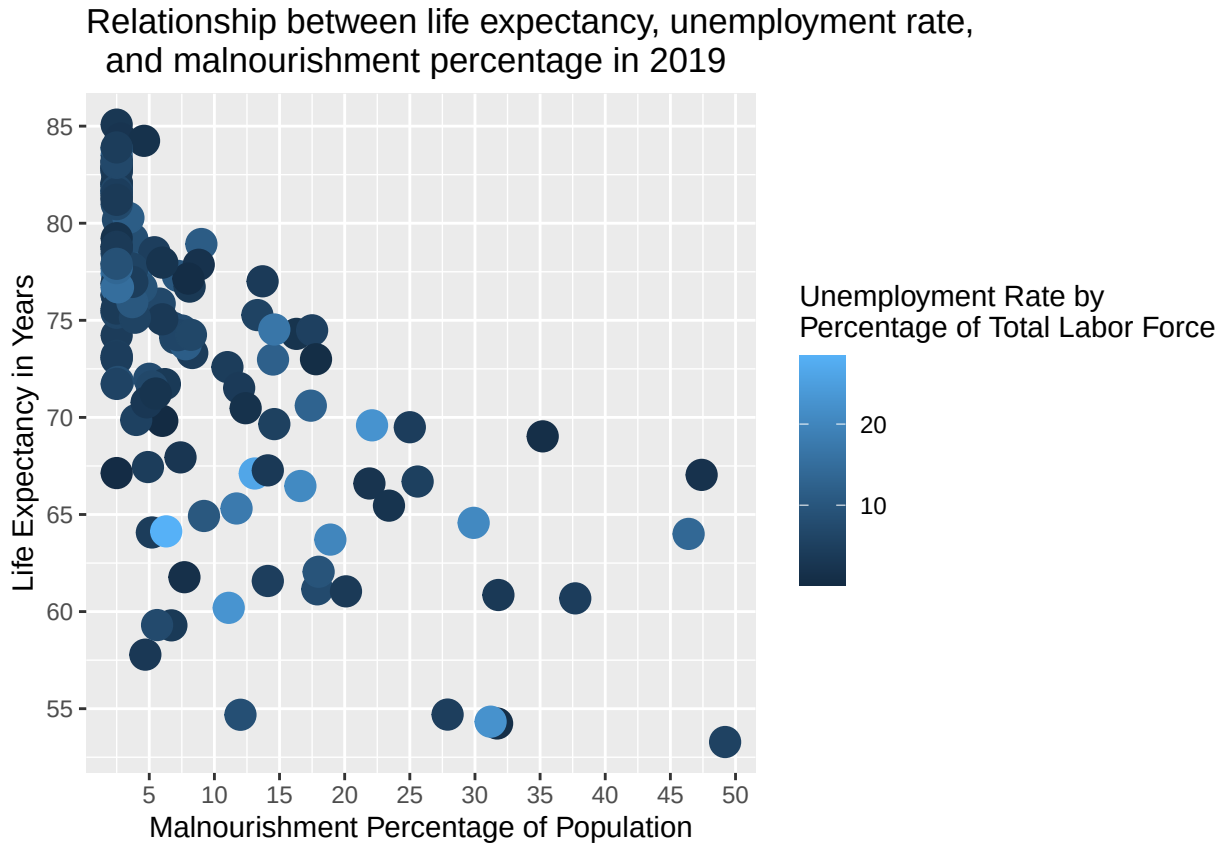



```
# Vis. 2-------------------------------------------------------------------------------
final_dataset %>%
  # map GNE and life expectancy to one another
  ggplot(aes(gne, life_expectancy, color = ..x..)) +
  geom_point( size = 2) +
  scale_x_log10() + # scale x-axis using log transform
  geom_smooth(color = "green") + #color geom_smooth line
  # add gradient color
  scale_color_gradient(low = "light green", high = "purple") +
  # add labs
   labs(title = "Life Expectancy vs. Gross National Expenditure (GNE)",
        subtitle = "(% of the Population vs. % of Total GDP)",
        x = "Log(Life Expectancy (%) of Population)",
        y = "Gross National Expenditure (% of Total GDP)",
        color = "GNE") +
  # add color and stylings
  theme_dark() +
```

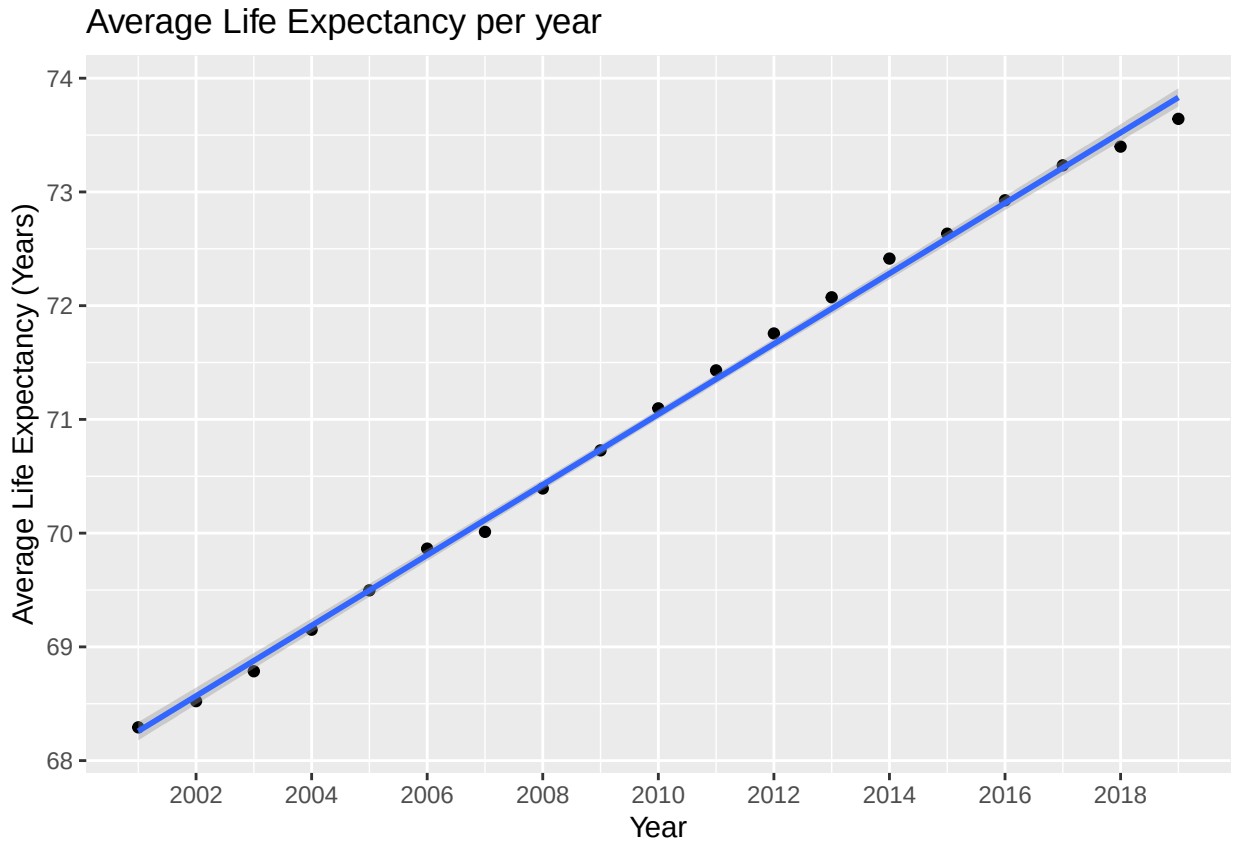
```
  theme(plot.background=element_rect(fill="#f1edf2"),
        legend.background = element_rect(colour="green",fill="white",
                                         size=.5, linetype="solid"),
        axis.text = element_text(colour = "#507d2a", face = "bold"),
        axis.title = element_text(colour = "#507d2a", face= "bold"),
        legend.text = element_text(colour = "#507d2a"),
        legend.title = element_text(colour = "#507d2a", face = "bold", size = 9),
        plot.title = element_text(colour = "#507d2a", face = "bold"),
        plot.subtitle = element_text(colour = "#507d2a"))
```



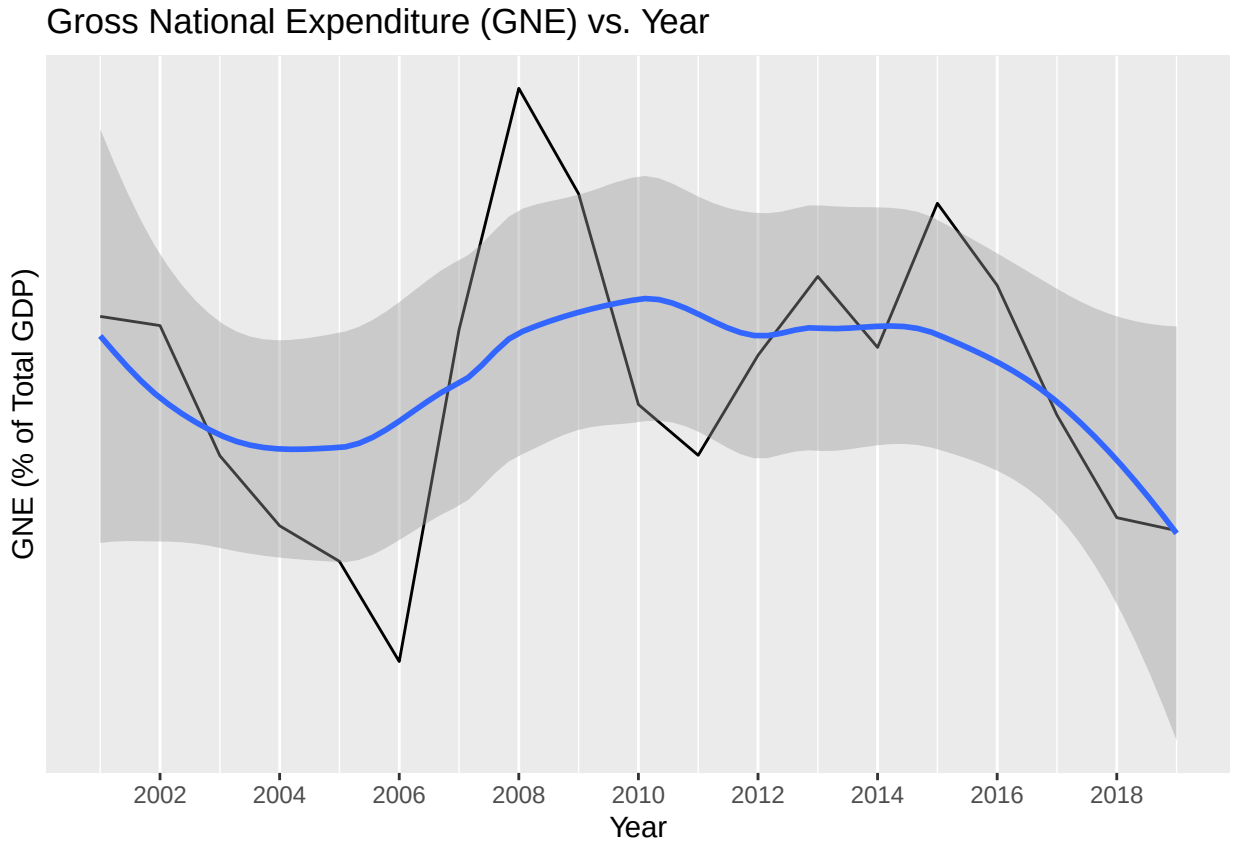

```
# Vis. 3-----------------------------------------------------------------------------
final_dataset %>%
  filter(year == 2019) %>%
  ggplot(aes(x = malnourish_perc, y = life_expectancy, color = unemploy_rate)) +
  geom_point(size = 5) +
  labs(title = "Relationship between life expectancy, unemployment rate,
  and malnourishment percentage in 2019", x = 'Malnourishment Percentage of Population', y = "Life Expe
Percentage of Total Labor Force") +
  scale_x_continuous(breaks = seq(0,50,5)) +
  scale_y_continuous(breaks = seq(40,100,5))
```



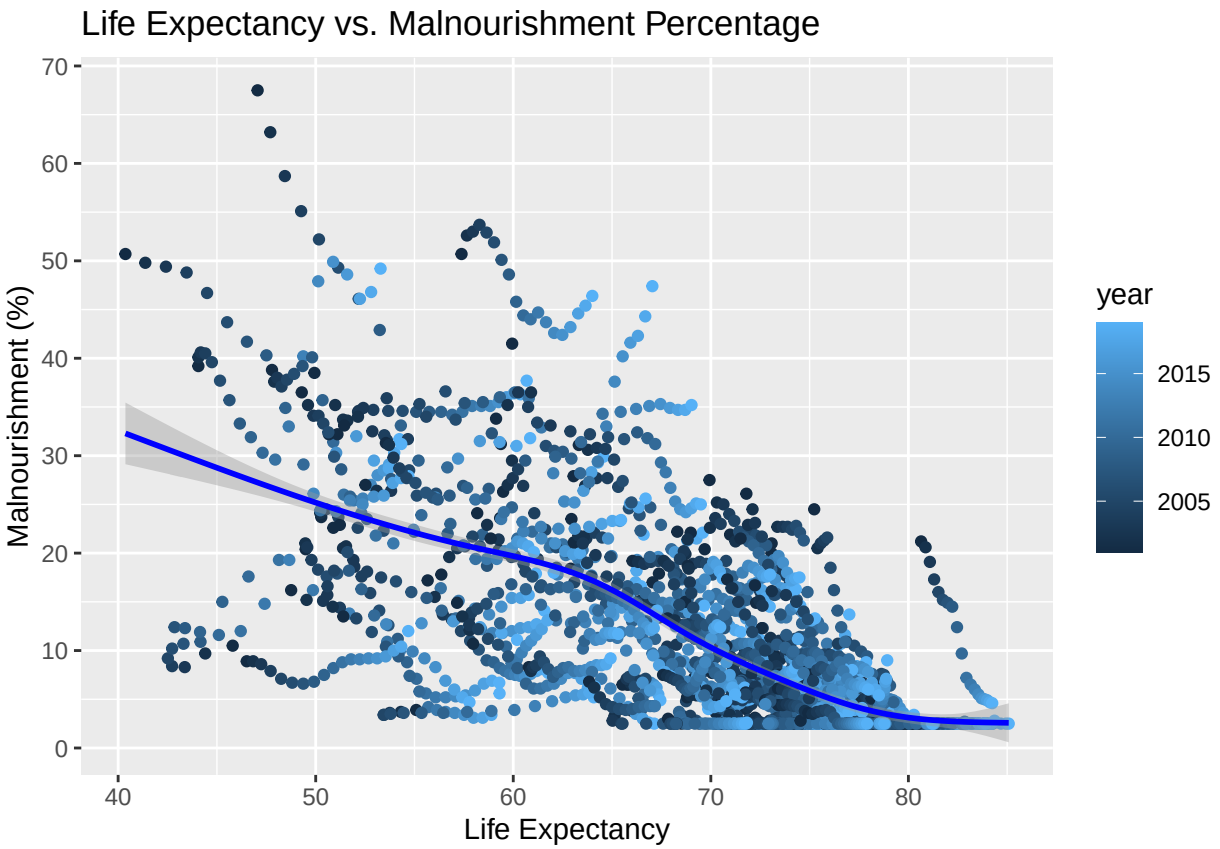

```
# Vis. 4-------------------------------------------------------------------------------
final_dataset %>%
  group_by(year) %>%
  summarize(avg_lifespan = mean(life_expectancy)) %>%
  ggplot(aes(x = year, y = avg_lifespan)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  labs(title = 'Average Life Expectancy per year', x = 'Year', y = 'Average Life Expectancy (Years)') +
  scale_x_continuous(breaks = seq(2000,2020,2)) +
  scale_y_continuous(breaks = seq(68,74,1))
```




```
# Vis. 5 ------------------------------------------------------------------------------
final_dataset %>%
  group_by(year) %>%
  summarize(gne = mean(gne)) %>%
  ggplot(aes(x=year,y=gne)) +
  geom_line() +
  geom_smooth() +
  labs(title = "Gross National Expenditure (GNE) vs. Year",
       x = "Year", y = "GNE (% of Total GDP)") +
  scale_x_continuous(breaks = seq(2000,2020,2)) +
  scale_y_continuous(breaks = seq(68,74,1))
```

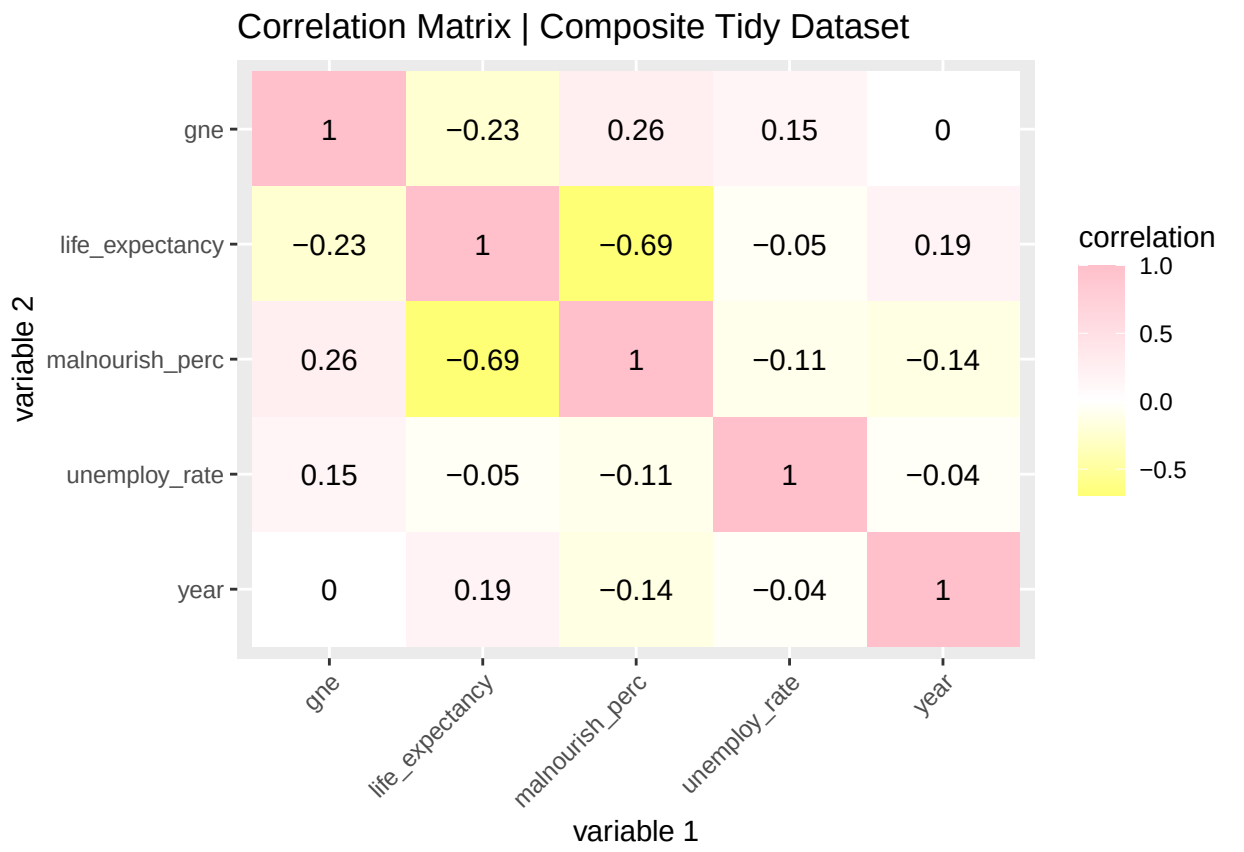



```
# Vis. 6 ------------------------------------------------------------------------------
final_dataset %>%
  ggplot(aes(x = life_expectancy, y = malnourish_perc, color = year)) +
  geom_point(stat = "identity") +
  geom_smooth(color = "blue") +
  labs(title = "Life Expectancy vs. Malnourishment Percentage",
       x = "Life Expectancy", y = "Malnourishment (%)") +
  scale_y_continuous(breaks = seq(0,70,10))
```




```
# Generating the correlation plot-------------------------------------------------------
numeric_only <- final_dataset %>%
  select_if(is.numeric)

cor(numeric_only, use = "pairwise.complete.obs") %>%
  # Save as a data frame
  as.data.frame %>%
  # Convert row names to an explicit variable
  rownames_to_column %>%
  # Pivot so that all correlations appear in the same column
  pivot_longer(-1,
               names_to = "other_var",
               values_to = "correlation") %>%
  # Define ggplot (reorder values on y-axis)
  ggplot(aes(x = rowname,
             y = ordered(other_var, levels = rev(sort(unique(other_var)))),
             fill = correlation)) +
  # Heat map with geom_tile
  geom_tile() +
  # Change the scale to make the middle appear neutral
  scale_fill_gradient2(low = "yellow", mid = "white", high = "pink") +
  # Overlay values
  geom_text(aes(label = round(correlation,2)), color = "black", size = 4) +
  # Angle the x-axis label to 45 degrees
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  # Give title and labels
```

```
  labs(title = "Correlation Matrix | Composite Tidy Dataset",
       x = "variable 1", y = "variable 2")
```

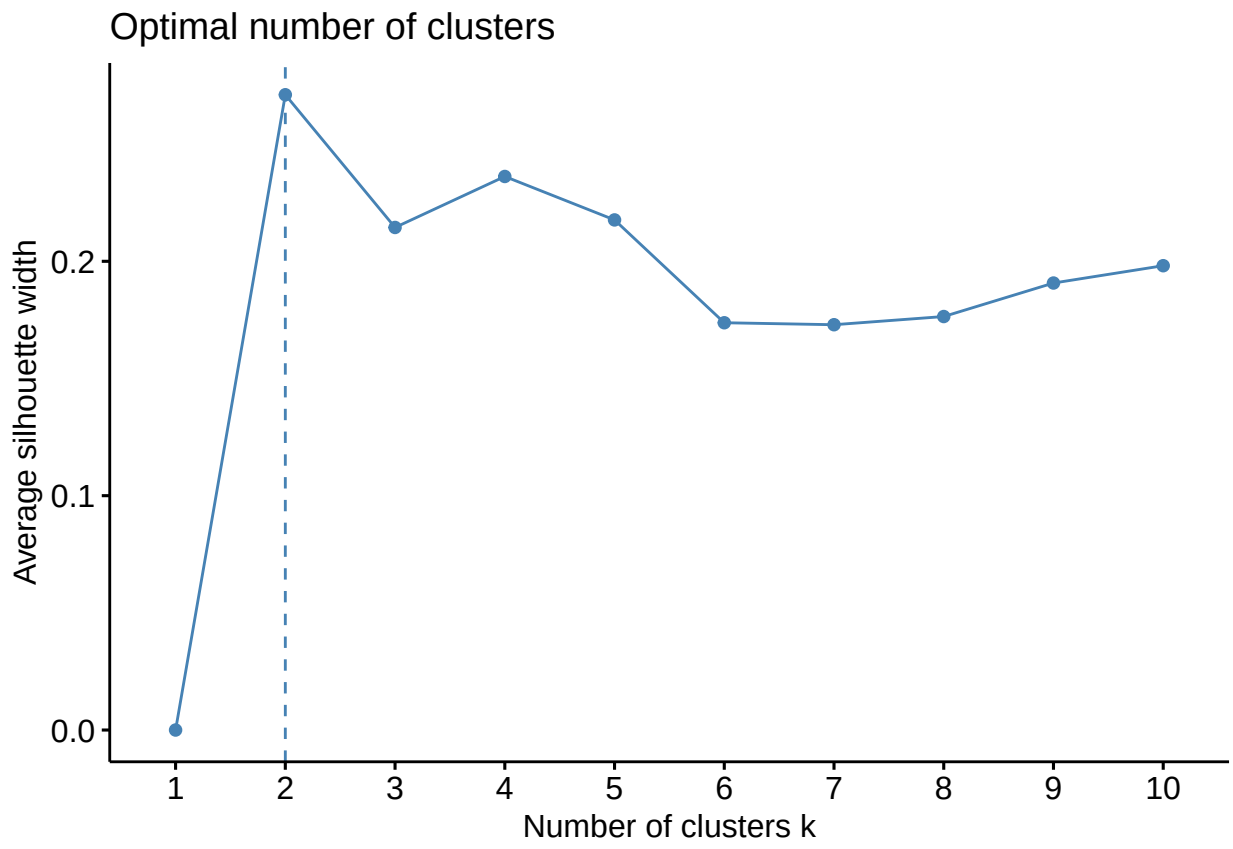


The most strongly correlated variables are malnourishment percentage and life expectancy, and this relationship is negative. GNE and unemployment rate share no correlation with the year variable. Life expectancy and GNE share a moderately correlated, negative relationship with one another and are reflected as such in the correlation plot. The visualization shows that the year and average life expectancy variables display a strong, positive linear relationship with one another despite the correlation matrix communicating that the correlation between these variables is minimal. The malnourishment percentage and unemployment rate did not share as significant a relationship as expected. In 2019, life expectancy, malnourishment, and unemployment percentage showed a moderately strong relationship that trends negatively. The relationship between life expectancy and malnourishment percentage was strong and linearly correlated. As life expectancy increased, the malnourishment percentage tended to decrease over time. There appeared to be no strong correlation between average GNE and year. However, there were notable fluctuations as time went on in the average GNE that resulted in it settling to a value lower than where it started.

Clustering

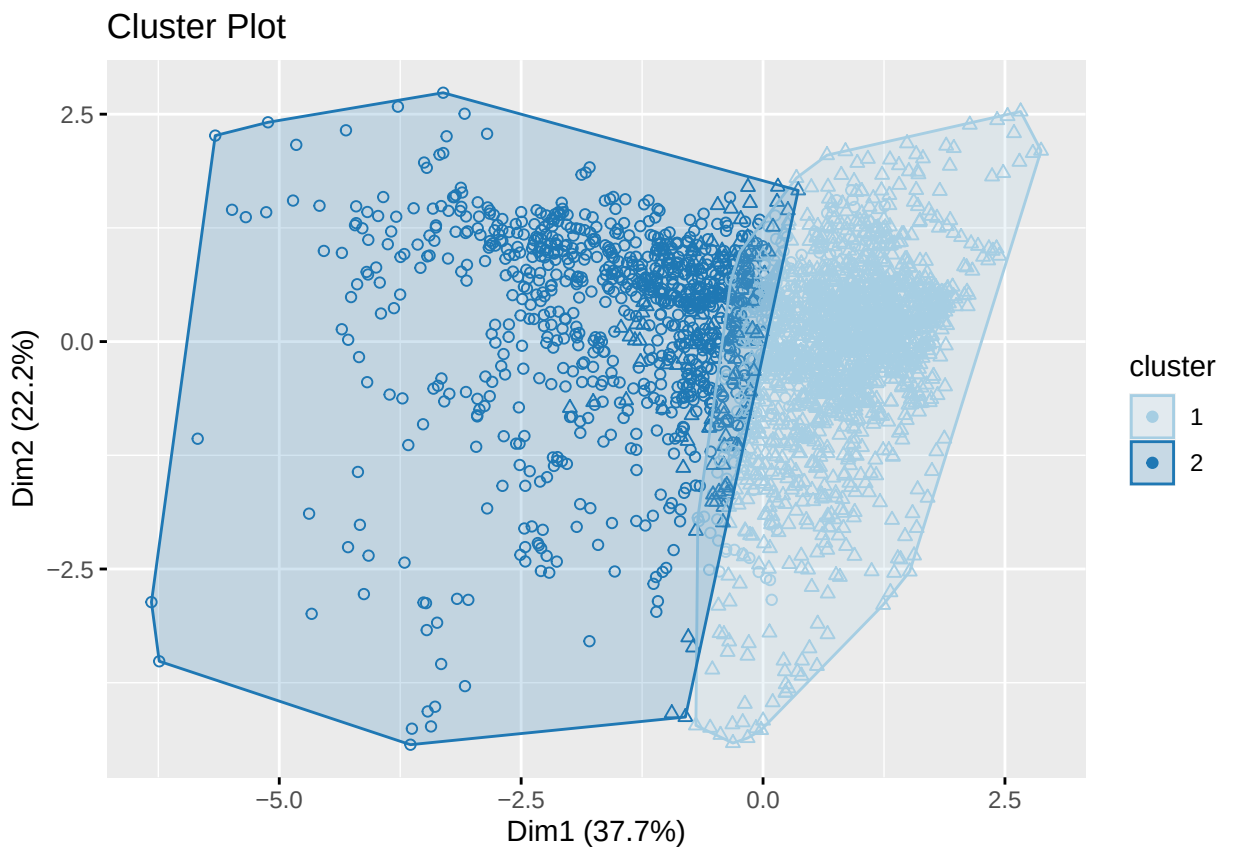
```
# Making the binary variable
# Determined by taking the average life expectancy for the year 2019 in our dataset
avg_life <- 71.04785

# Add binary variable classifying a country's life expectancy as
# Below or Above avg. based on the avg. life expectancy calculated from
```

```r
# entries corresponding to 2019
final_dataset <- final_dataset %>%
  mutate(above_avg_lifespan = ifelse(final_dataset$life_expectancy > avg_life, 1, 0))

# Prepare data: Selecting numeric variables and scale them
final_scaled <- final_dataset %>%
  select_if(is.numeric) %>%
  select(-above_avg_lifespan) %>%
  scale

# Determine the optimal number of clusters
cluster_viz <- fviz_nbclust(final_scaled, pam, method = "silhouette")
cluster_viz
```




```r
# Apply the PAM clustering algorithm
pam_results <- final_scaled %>%
  pam(k = 2)

# Save cluster assignment as a column in our dataset
final_pam <- final_dataset %>%
  mutate(cluster = as.factor(pam_results$clustering))

#Visualizing the clusters
fviz_cluster(pam_results, data = final_dataset,
             shape = as.factor(final_dataset$above_avg_lifespan),
             palette = "Paired", geom = "point") +
```

```
  guides(shape = guide_legend(title = "shape")) +
  ggtitle(label = "Cluster Plot")
```




```
# Observe the average silhouette width --> 0.2710
pam_results$silinfo$avg.width
```

```
## [1] 0.2710735
```

```
# Determine the medoids at the center of each cluster
pam_results$medoids
```

```
##            year        gne unemploy_rate life_expectancy malnourish_perc
## [1,]  0.1814206 -0.3044639    -0.1421260       0.4907740      -0.6943843
## [2,] -0.5505154  0.2575885    -0.4045916      -0.6368158       0.7628814
```

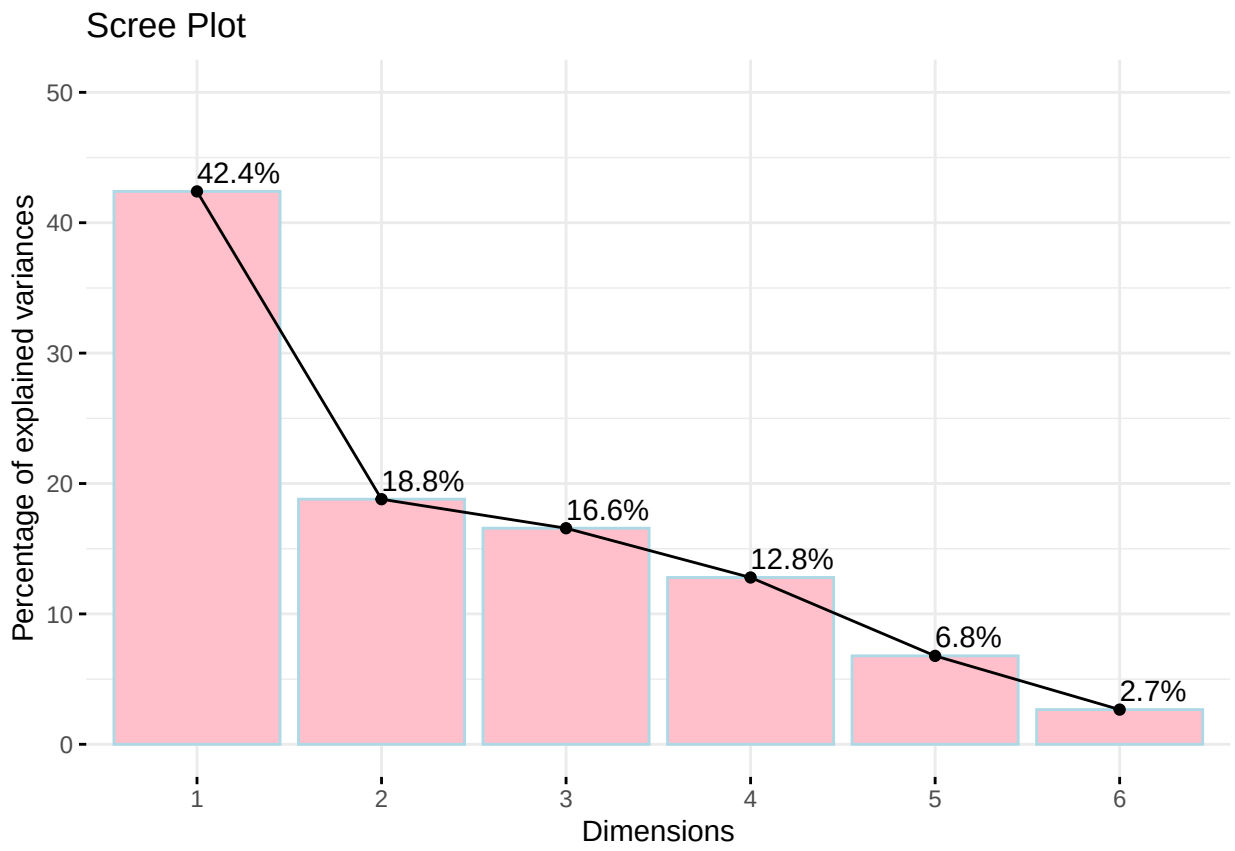
The optimal number of clusters is two to maximize silhouette width. The average silhouette width is 0.2710, a weak structure that suggests that our clusters are not well separated and that observations belonging to one cluster are not that distant from the observations residing in the closest neighboring cluster. The medoid at the center of cluster 1 has the following values: year: 0.1814206, gne: -0.3044639, unemploy_rate: -0.1421260, life_expectancy: 0.4907740, and malnourish_perc: -0.6943843. The medoid at the center of cluster 2 has the following values: year: -0.5505154, gne: 0.2575885, unemploy_rate: -0.4045916, life_expectancy: -0.6368158, and malnourish_perc: 0.7628814.

Dimensionality Reduction | PCA

```
# Select the numeric columns in the dataset
final_reduced <- final_dataset %>%
  select_if(is.numeric)

# Perform PCA on all numeric variables in the dataset
pca <- final_reduced %>%
  select_if(is.numeric) %>%
  scale %>%
  prcomp

# View PC's
pca
```

```
## Standard deviations (1, .., p=6):
## [1] 1.5950015 1.0619460 0.9971216 0.8760513 0.6378714 0.3995544
##
## Rotation (n x k) = (6 x 6):
##                              PC1          PC2         PC3        PC4         PC5
## year                -0.158356233  0.006014414 -0.91614994  0.3614690  0.04921915
## gne                  0.244002820  0.556828937 -0.32505163 -0.7072773 -0.15567780
## unemploy_rate       -0.007503874  0.814461131  0.20846507  0.5067474  0.15171704
## life_expectancy     -0.580227020  0.009230498 -0.02465204 -0.2301573  0.13981907
## malnourish_perc      0.525734307 -0.120640609 -0.08374018 -0.0572400  0.80291501
## above_avg_lifespan -0.549790563  0.109174494  0.06271286 -0.2367641  0.53488499
##                             PC6
## year                 0.04974019
## gne                  0.01639463
## unemploy_rate       -0.11544730
## life_expectancy     -0.76819395
## malnourish_perc     -0.23256813
## above_avg_lifespan   0.58285433
```

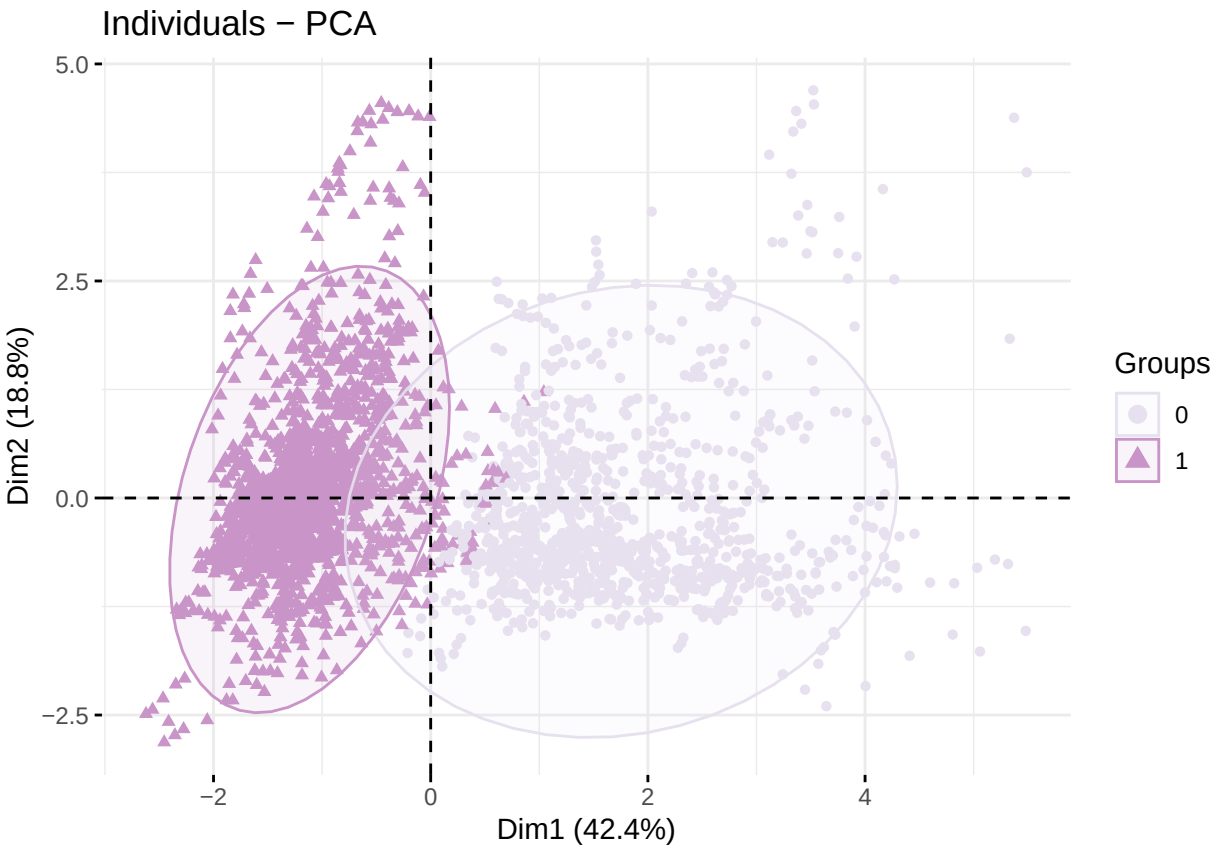
```
# Create a scree plot to look at the percent variation explained by the PCs
fviz_eig(pca, addlabels = TRUE,
         ylim = c(0, 50), barcolor = "light blue",
         barfill = "pink") +
  ggtitle(label = 'Scree Plot')
```




```
# Observe the average silhouette width --> 0.2710
pam_results$silinfo$avg.width
```

```
## [1] 0.2710735
```

```
# Represent the clusters on PC1 and PC2
fviz_pca_ind(pca, label="none", habillage=final_dataset$above_avg_lifespan,
     addEllipses=TRUE, ellipse.level=0.95, palette = "PuRd")
```



The first and second principal components explain 61.2 percent of the total variation in our data. Of the original variables in our dataset, the first principal component shows the strongest correlation to a country's life expectancy and malnourishment percentage and a weak correlation to year, GNE, and unemployment rate. The entry corresponding to the malnourishment percentage is positive. In contrast, the value corresponding to life expectancy is negative, indicating that these variables are correlated but have opposite effects on one another. Scoring high on the first principal component indicates that a country's life expectancy will be lower as its malnourishment rate increases and can be considered a measure of how long people's lives in a country are as a function of how many people are malnourished. The second principal component strongly correlates with a country's GNE and unemployment rate and is weakly correlated to year, life expectancy, and malnourishment percentage. These values are both positive, indicating that these metrics tend to change positively together; therefore, a country with a higher GNE also tends to have a higher percentage of unemployment. Cluster 1 had a negative PC1 value and a positive PC2 value indicating that countries in this cluster tended to have a higher life expectancy, lower malnourishment percentage, higher GNE, and a higher unemployment rate. Cluster 2 had a positive PC1 value and a positive PC2 value indicating that countries in this cluster tended to have higher lower life expectancies, higher malnourishment rates, higher GNE, and a higher unemployment rate.

Classification and Cross Validation

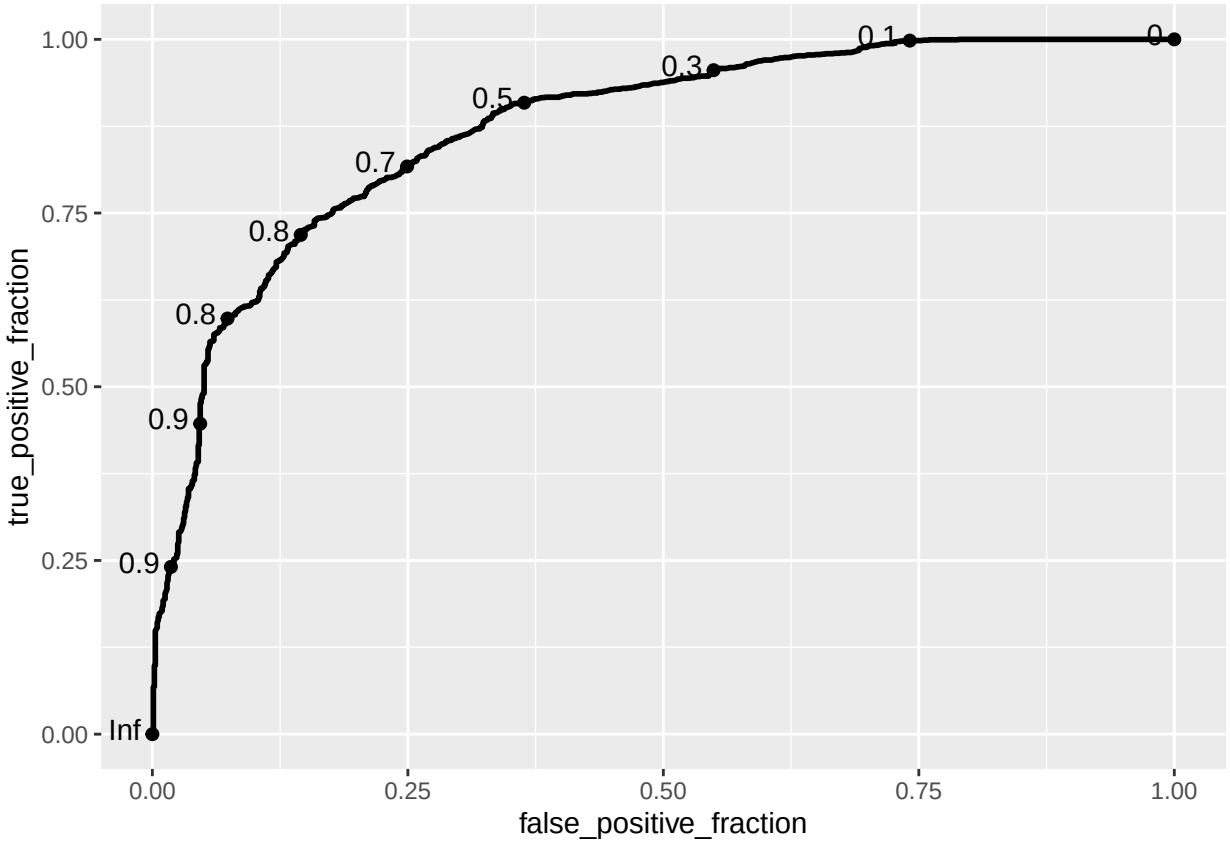
Logistic Regression Model

```
# Logistic Regression Model
fit_log <- glm(above_avg_lifespan ~ unemploy_rate + malnourish_perc,
               data = final_dataset, family = "binomial")
```

```r
# View model summary
summary(fit_log)
```

```
##
## Call:
## glm(formula = above_avg_lifespan ~ unemploy_rate + malnourish_perc,
##     family = "binomial", data = final_dataset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0411  -0.5596   0.5247   0.6031   2.5738
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.498975   0.115098  21.712   <2e-16 ***
## unemploy_rate   -0.007706   0.008158  -0.945    0.345
## malnourish_perc -0.218006   0.009048 -24.094   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3598.0  on 2691  degrees of freedom
## Residual deviance: 2396.3  on 2689  degrees of freedom
## AIC: 2402.3
##
## Number of Fisher Scoring iterations: 5
```

```r
# Calculate the predicted values using our log. reg. model
final_pred <- final_dataset %>%
  mutate(prediction = predict(fit_log, type = 'response'))

# Plotting the ROC Curve
ROC <- ggplot(final_pred) +
  geom_roc(aes(d = above_avg_lifespan,
               m = prediction),
               n.cuts = 10)

# Visualizing ROC curve
ROC
```




```
# Calculating AUC to assess our model accuracy --> 0.8629
calc_auc(ROC)$AUC
```

```
## [1] 0.872006
```

K - Fold Cross Validation

```
# Choose number of folds
k = 10

final_reduced <- final_dataset %>%
  select_if(is.numeric) %>%
  select(-life_expectancy)

# Randomly order rows in the dataset
data <- final_reduced[sample(nrow(final_reduced)), ]

# Create k folds from the dataset
folds <- cut(seq(1:nrow(data)), breaks = k, labels = FALSE)

# Initialize a vector to keep track of the performance
perf_k <- NULL

# Use a for loop to get diagnostics for each test set
for(i in 1:k){
```

```r
  # Create train and test sets
  train <- data[folds != i, ] # all observations except in fold i
  test <- data[folds == i, ]  # observations in fold i

  # Train model on train set (all but fold i)
  final_log <- glm(above_avg_lifespan ~ ., data = train, family = "binomial")

  # Test model on test set (fold i)
  df <- data.frame(
    predictions = predict(final_log, newdata = test, type = "response"),
    above_avg_lifespan = test$above_avg_lifespan)

  # Consider the ROC curve for the test dataset
  ROC <- ggplot(df) +
    geom_roc(aes(d = above_avg_lifespan, m = predictions))

  # Get diagnostics for fold i (AUC)
  perf_k[i] <- calc_auc(ROC)$AUC
}

mean(perf_k) # --> 0.8727
```

```
## [1] 0.8727329
```

The average performance across 10 k folds was 0.8727, indicating that our model could predict whether a country's life expectancy could be classified as above or below average when using all of the original variables, excluding the life expectancy and the categorical variables. This is comparable to the AUC value we received from the ROC curve generated from our linear model that predicts whether a country's life expectancy is considered above average using only the malnourishment rate and unemployment percentage to predict it. Furthermore, there are no signs of overfitting, as k-fold cross-validation revealed that our model's accuracy in predicting the unseen test data was slightly higher on average than the accuracy measured on the model trained on the full dataset. Interestingly, the model used for k-fold cross-validation included all of the original variables, excluding life expectancy, and obtained an accuracy that rivaled that of the logistic regression model that only used two variables to predict whether a country's life expectancy was above average. This suggests that our dataset can obtain a high classification accuracy without an overcomplicated model.

Acknowledgements:

Team Contributions:

- Jordan Toler: Full Effort
- Tim Giang: Full Effort
- Elizabeth Lam: Full Effort

Citations:

Style: MLA9

"Dictionary.net." *Dictionary.net | Find Definitions and Meanings of Words*, https://www.dictionary.net/.

Roser, Max, et al. “Life Expectancy.” *Our World in Data*, 23 May 2013, https://ourworldindata.org/life-expectancy.

“World Development Indicators.” *DataBank*, 16 Sept. 2022, https://databank.worldbank.org/source/world-development-indicators.

» The World Development Indicators database was used to manually create the customized datasets that made our project possible.