Social And Behavioral Factors Affecting COVID-19 Transmissibility

Nicholas Pao

4/27

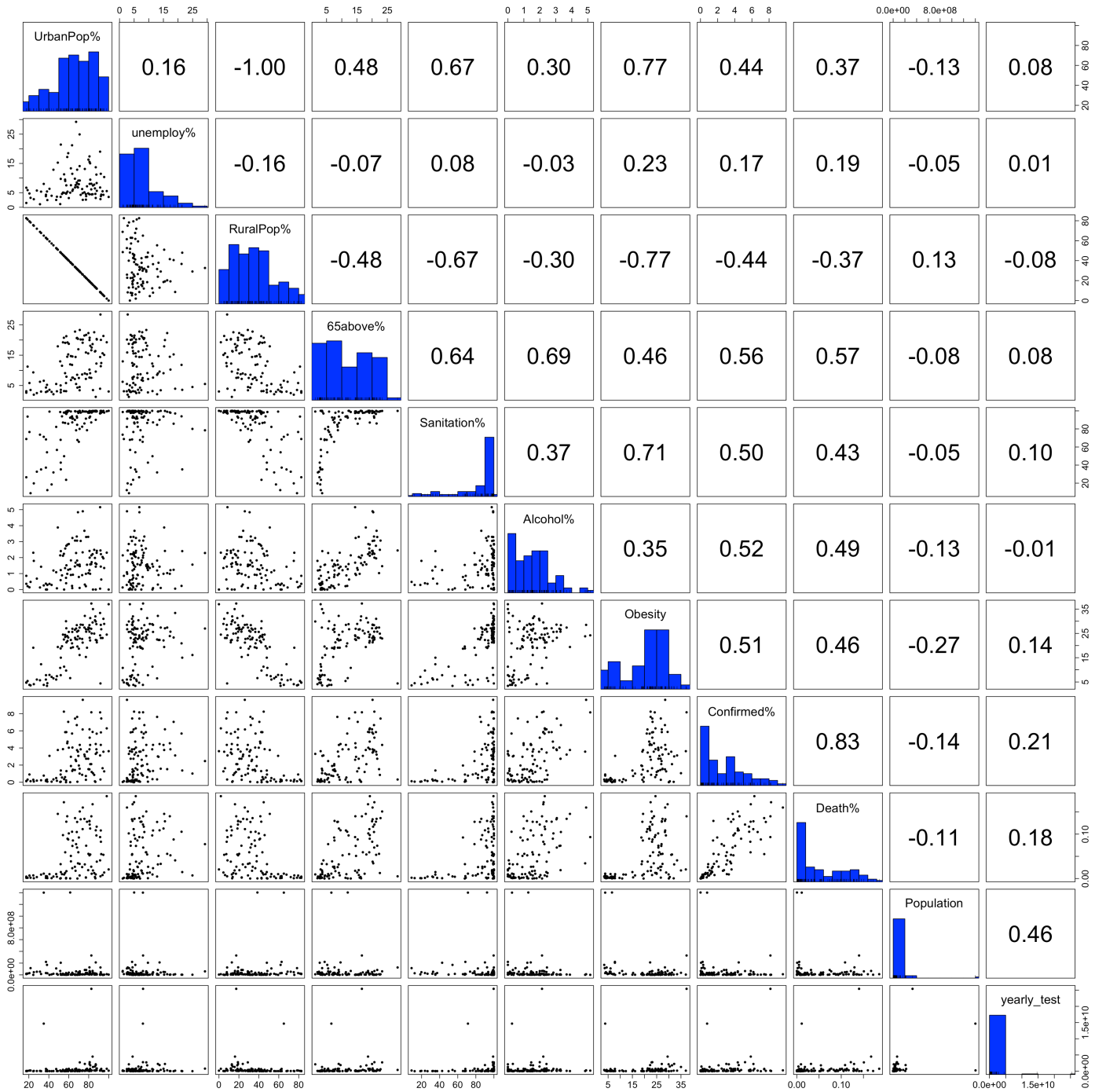
Title and Introduction

The main purpose of this project is to evaluate the impact of social and behavioral factors on COVID-19 transmissibility. In order to accomplish this, we looked at the 'covid2020' dataset which is a merged dataset composed of the 'countryhealthstats' dataset from the World Bank website, 'diet2020' dataset from Kaggle, and 'totaltests' dataset from the 'Our World in Data' website. The variables contained in the 'covid2020' dataset are the % of urban population, the % of unemployment, the % of rural population, the % of people above the age of 65, the % of people with access to proper sanitation, alcohol intake as a percentage of total calories in diet, the % of obesity,the % of population with confirmed covid cases, the % of population dead due to COVID-19, the population, and the total yearly tests for different countries around the world. This dataset also has 102 observations with each observation representing a unique country. The 'covid2020' dataset is particularly interesting to us because social and behavioral factors described in this dataset shape the everyday life of people and thus will no doubt impact the disease prevalence of diseases like COVID-19 around the world. Some interesting trends we expect are a positive correlation between the percentage of elderly population in a country and the death due to COVID-19 rate, and we also expect health conditions like obesity and undernourishment to be positively correlated with COVID-19 occurrence. The 'countryhealthstats' dataset was tidied before it merged with the other two datasets to create the 'covid2020' dataset. In the 'countryhealthstats' dataset, we moved the different variables from the 'Series Name' column to separate columns of their own using pivot_wider, thus tidying the dataset. In this project, we also explore the dataset using PCA and clustering to view the different ways in which the countries are grouped or related. We also import a new dataset which reports the HDI of each country from the World Population Review Website and compare to see if the country clusters match the development status of the countries. Finally, we use classification to see if we can use the COVID-19 related variables in the dataset to predict the development status of the country and then we evaluate that classification model to check if it has any signs of overfitting.

Correlation Matrix with Univariate and Bivariate

graphs

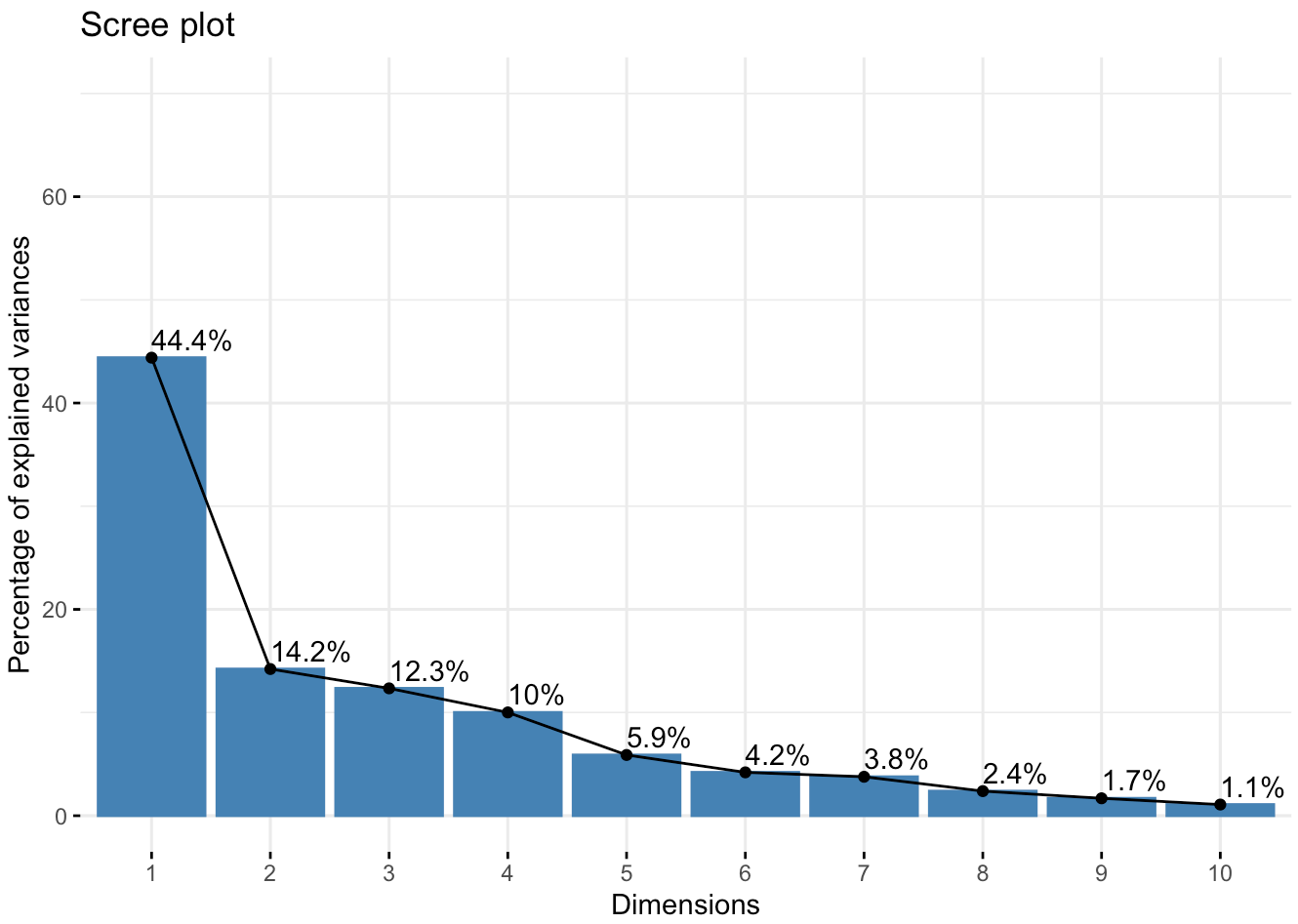
```
#Call all the relevant packages
library(tidyverse)
library(cluster)
library(factoextra)
library(GGally)
library(ggplot2)
library(plotROC)
library(caret)
library(psych)

# Call the 'covid2020' dataset
library(readr)
covid2020 <- read_csv("covid2020.csv")


#Remove the variable 'Handwash%' from the 'covid2020' dataset as it contains too many 'N
A' values
#covid2020 <- covid2020 %>% select(-`Handwash%`, -Undernourished)

# Save a new dataset called 'covid2020num' containing only the numeric variables from th
e 'covid2020' dataset
covid2020num <- covid2020 %>% select(is.numeric)



# Make a correlation matrix with bivariate and univariate graphs for the 'covid2020num'
dataset
pairs.panels(covid2020num,
             method = "pearson", # correlation coefficient method
             hist.col = "blue", # color of histogram
             smooth = FALSE, density = FALSE, ellipses = FALSE, cex.labels=2, cex.axis=
1.5)
```



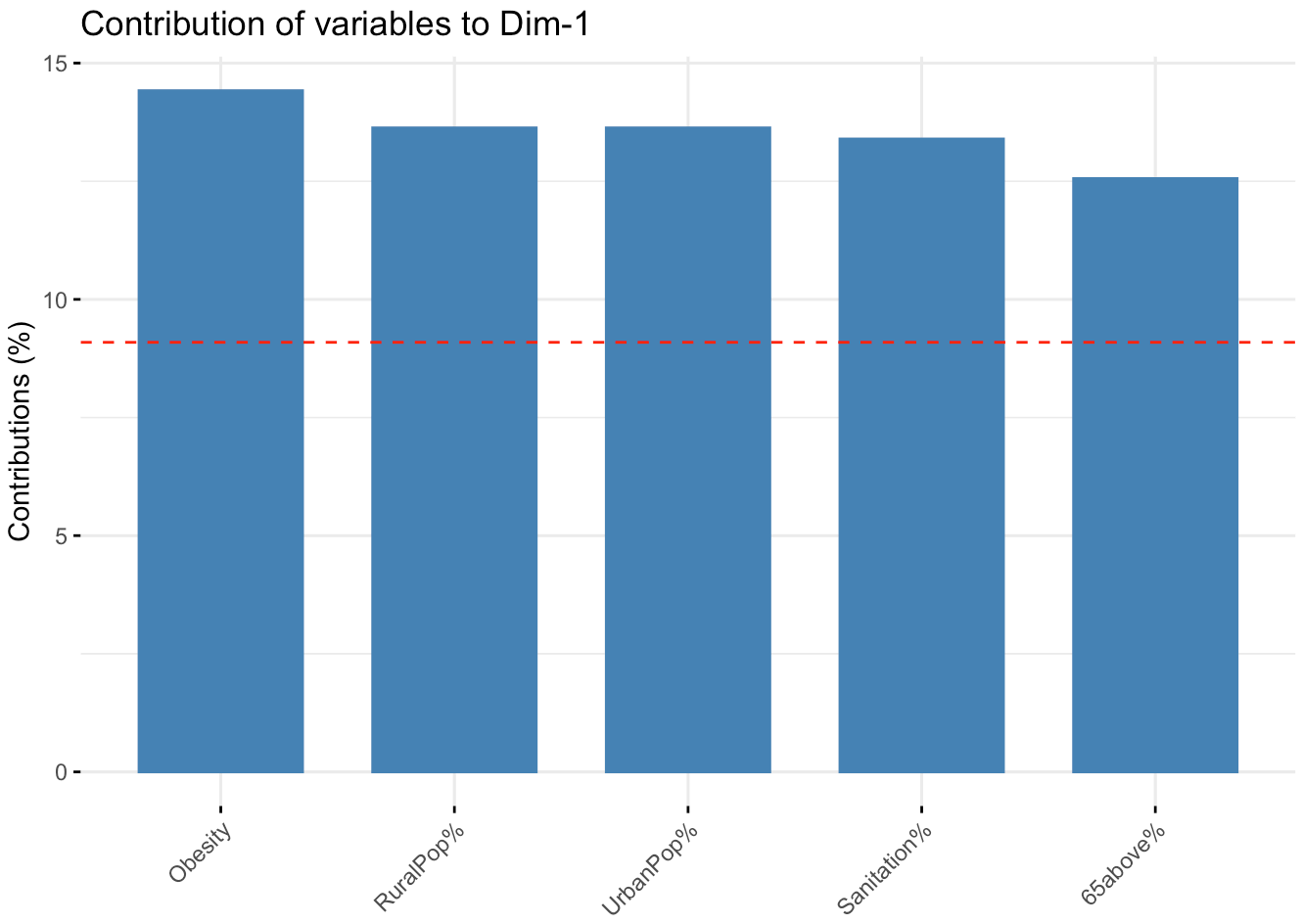
The needed packages were called then our covid2020 dataset, used from our last project, was uploaded using the read_csv function. The covid2020 dataset included the country names and statistics about their population such as the urban population %, the confirmed cases, %, the amount of yearly tests, etc. We then selected only the numeric variables from our covid2020 dataset and saved it to covid2020num and made a correlation matrix with bivariate and univariate graphs for its respective numeric variables. Technically, the UrbanPop% and the RuralPop% were the most correlated with a correlation coefficient of -1.00 but this was expected as they are complete opposite statistics. The confirmed case and death percentage were the second most correlated with a correlation coefficient of about .83 which makes sense as the death percentage due to covid19 is sure to be in high correspondence among those who had caught the virus with the lethality of covid19 around the world. Unemployment% and yearly_tests as well as Alcohol% and yearly_tests were tied for the least correlated pairs of variables with correlation coefficients of 0.01 and -0.01 respectively. This makes sense as one would expect these variables to have no relationship with each other.

PCA

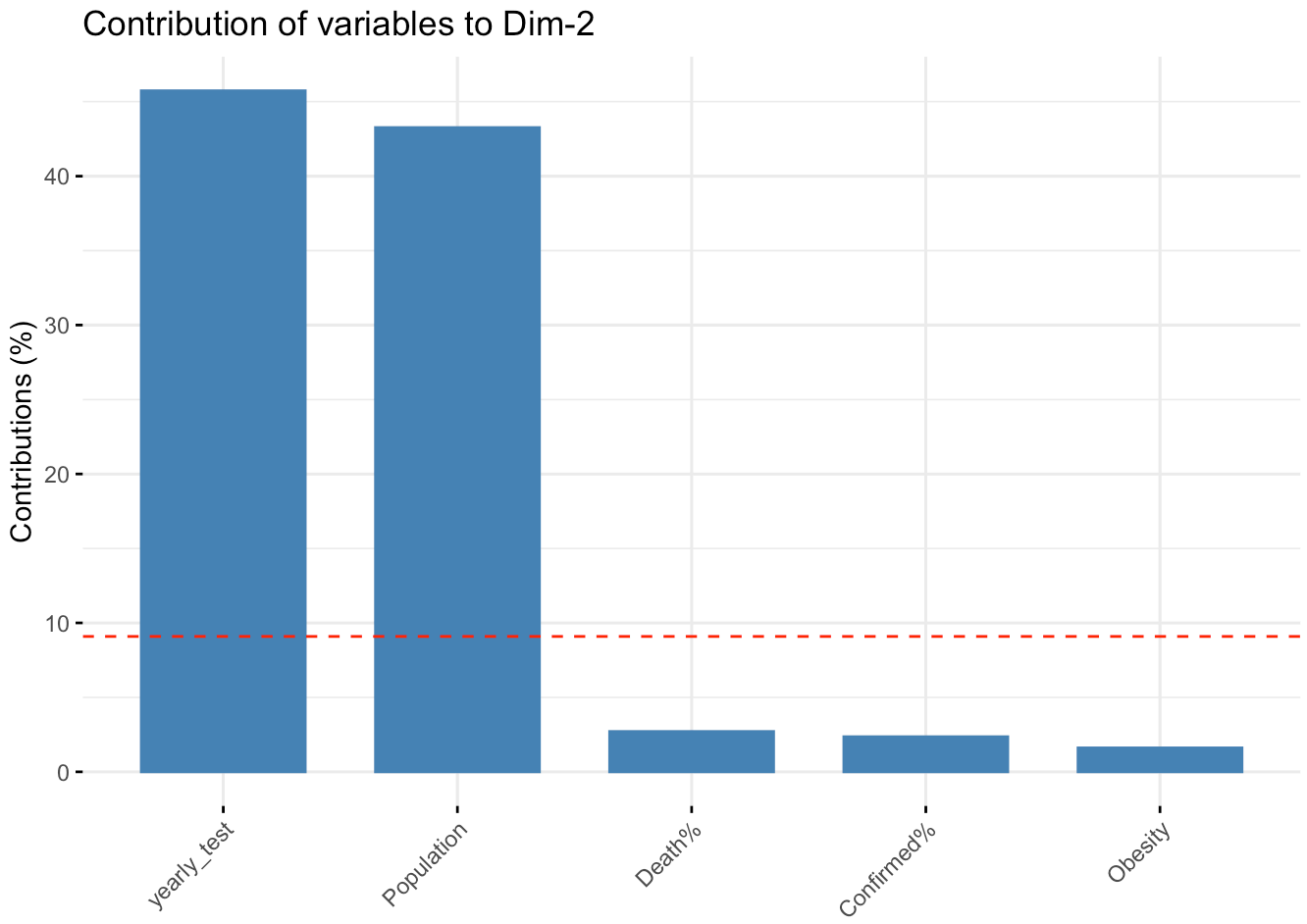
```
#Make a 'covid2020scaled' dataset containing only scaled numeric variables with no missi
ng values from the 'covid2020num' dataset
covid2020scaled <- covid2020num %>% scale %>% as.data.frame() %>% na.omit

#Conduct a pca on the 'covid2020scaled' dataset
pca <- covid2020scaled %>% prcomp()

# Visualize percentage of variances for each PC in a scree plot
fviz_eig(pca, addlabels = TRUE, ylim = c(0, 70))
```




```
# Visualize the 5 top contributions of the variables to the PCs as a percentage
  # Note the red dash line indicates the average contribution
fviz_contrib(pca, choice = "var", axes = 1, top = 5) # on PC1
```



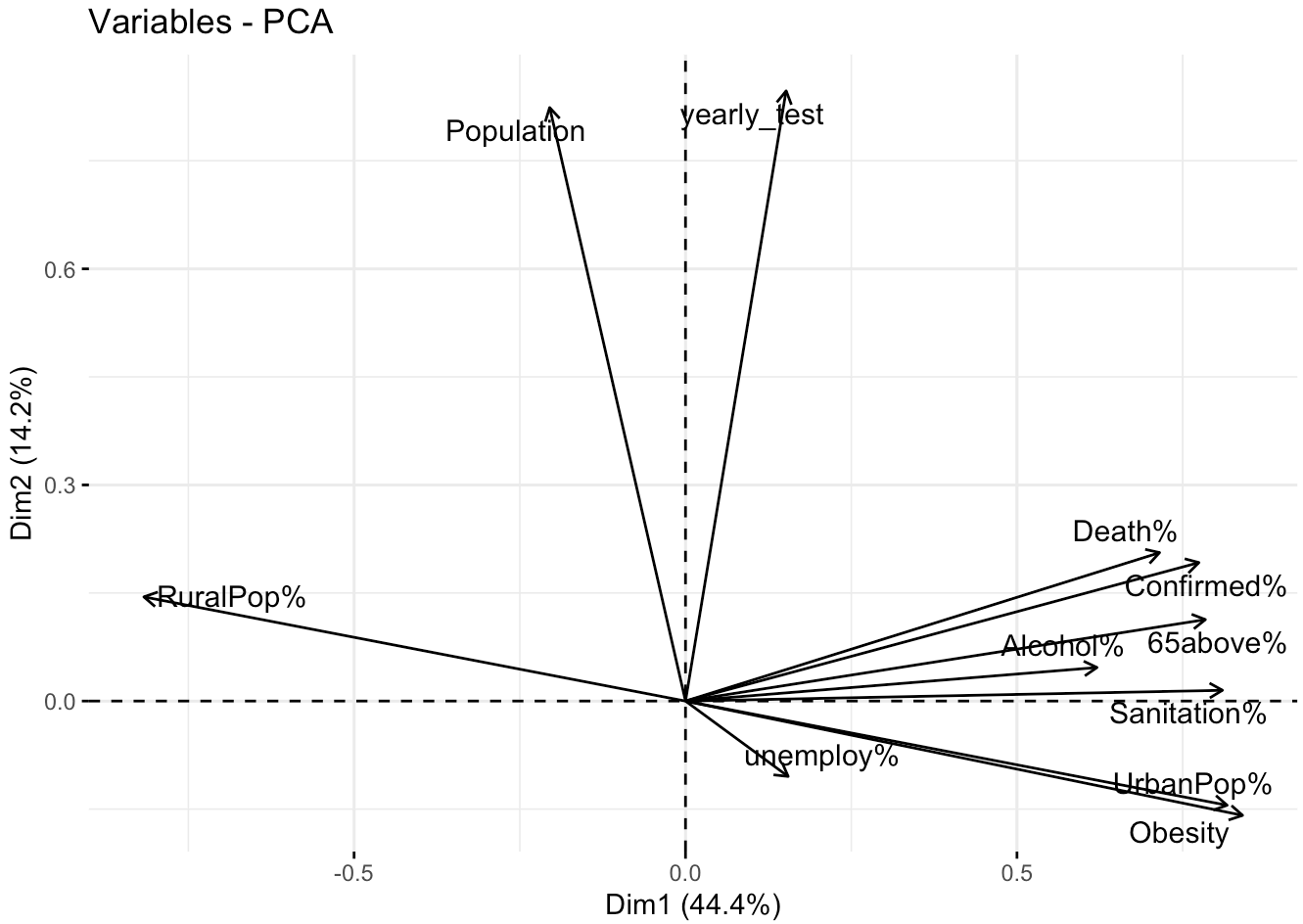

```
fviz_contrib(pca, choice = "var", axes = 2, top = 5) # on PC2
```



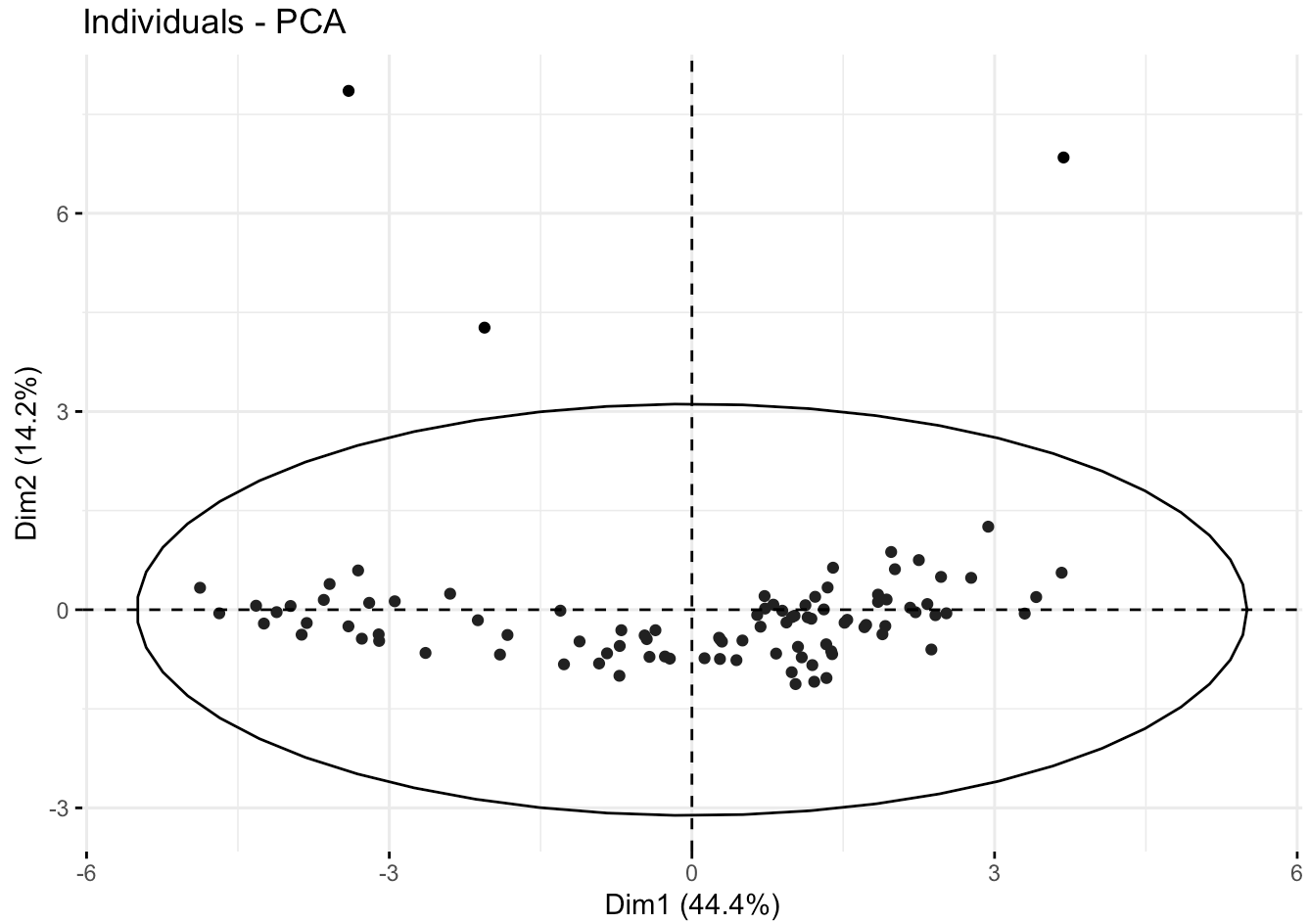

```
# Visualize the contributions of the variables to the PCs in a table
get_pca_var(pca)$coord
```

```
##                  Dim.1       Dim.2      Dim.3       Dim.4        Dim.5
## UrbanPop%    0.8170718 -0.14462983 -0.3806190 -0.21671324 -0.002947619
## unemploy%    0.1548693 -0.10463804 -0.4650332  0.73889644 -0.404452165
## RuralPop%   -0.8170718  0.14462983  0.3806190  0.21671324  0.002947619
## 65above%     0.7842379  0.11313034  0.3795558 -0.21724387 -0.227608461
## Sanitation%  0.8099983  0.01499984 -0.1601688 -0.21152843 -0.107625245
## Alcohol%     0.6207213  0.04661748  0.5426893 -0.08681380 -0.252557176
## Obesity      0.8403316 -0.15867501 -0.3156407 -0.01270116  0.191956365
## Confirmed%   0.7744871  0.19227217  0.2874412  0.38868273  0.141004360
## Death%       0.7151574  0.20610430  0.3131063  0.41094933  0.115949654
## Population  -0.2049740  0.82385372 -0.2230074 -0.20322946 -0.399420518
## yearly_test  0.1518711  0.84705073 -0.2488066  0.06187560  0.359456007
##                  Dim.6       Dim.7       Dim.8        Dim.9       Dim.10
## UrbanPop%    0.12139866 -0.22401026  0.08160311 -0.055193464  0.030200243
## unemploy%    0.10965153  0.13637576  0.05511899 -0.014218089  0.004022261
## RuralPop%   -0.12139866  0.22401026 -0.08160311  0.055193464 -0.030200243
## 65above%    -0.02691828  0.19237524  0.32250542  0.050700522 -0.128714051
## Sanitation% -0.37677305  0.28912852 -0.11189648 -0.108366840  0.143161087
## Alcohol%     0.40478463  0.05007399 -0.21859071 -0.006788501  0.083001770
## Obesity      0.01943614  0.14086609 -0.21222879  0.237516604 -0.149680619
## Confirmed%  -0.12286424 -0.14551136 -0.10619581 -0.239396117 -0.151565069
## Death%      -0.17146028 -0.21204544  0.07832926  0.212131965  0.149452969
## Population  -0.12323793 -0.20433772 -0.11483954  0.069828031 -0.059977823
## yearly_test  0.23610434  0.21653469  0.08259900 -0.045781256  0.049232654
##                  Dim.11
## UrbanPop%   2.016544e-16
## unemploy%   0.000000e+00
## RuralPop%   2.016544e-16
## 65above%    7.915399e-33
## Sanitation% 1.583080e-32
## Alcohol%    3.957700e-32
## Obesity     3.166160e-32
## Confirmed%  3.957700e-33
## Death%      3.957700e-32
## Population  5.936550e-33
## yearly_test 1.088367e-32
```

```
# Visualize the contributions of the variables to the PCs in a correlation circle
fviz_pca_var(pca, col.var = "black",
             repel = TRUE) # Avoid text overlapping
```




```
# Visualize the individuals according to PC1 and PC2
fviz_pca_ind(pca,
             geom.ind = "point", # show points only (nbut not "text")
             palette = c("#00AFBB", "#E7B800", "#FC4E07"),
             addEllipses = TRUE, # Concentration ellipses
             legend.title = "outcome"
             )
```




```
#Total Variation Explained by the 2 PCs
44.4+14.2
```

```
## [1] 58.6
```

In the code chunk, we first scaled the 'covid2020num' dataset and removed missing values to ready it for PCA analysis. Then, we conducted PCA on the scaled dataset. We visualized the percentage of variance explained by each PC. The first PC explained about 44.4% of the variation, the second PC explained about 14.2% of the variation, the third PC explains about 12.3% of the variation, and the fourth PC explains about 10% of the variation. Technically, we would need to include 4 dimensions to have at least 80% variance explained, but for visualization purposes we will be sticking 2 principal components since that is easy to visualize. We then visualized which were the top 5 variables contributing to each of the 2 principal components. The top 5 contributing variables to the first principle component are 'Obesity', 'RuralPop%', 'UrbanPop%', 'Sanitation%', and '65above%', and all these 5 variables contributed around an equal amount to PC1 ranging from 10-15% contribution per variable. The top 5 contributing variables to the second principle component are 'yearly_test', 'population', 'Death%', 'Confirmed%', and 'Obesity', and here only 'yearly_test' and 'Population' contributed significantly to PC2 with each variable contributing around 40%, while the other 3 variables each had contributions below 5% to PC2. Based on the values of how each variable contributed to each PC, we determined what it would mean to score high in each PC. If a country were to score high in PC1, it would mean that the country has a high urban population %, a moderately high unemployment rate, a low rural population %, a high population of people above 65 years old, a high availability of sanitation facilities, a relatively high alcohol consumption %, a high obesity rate, a high % of COVID-19 confirmed cases, a high percentage of COVID deaths, a moderately low population, and moderate amount of COVID-19 tests conducted yearly compared to other countries. If a country

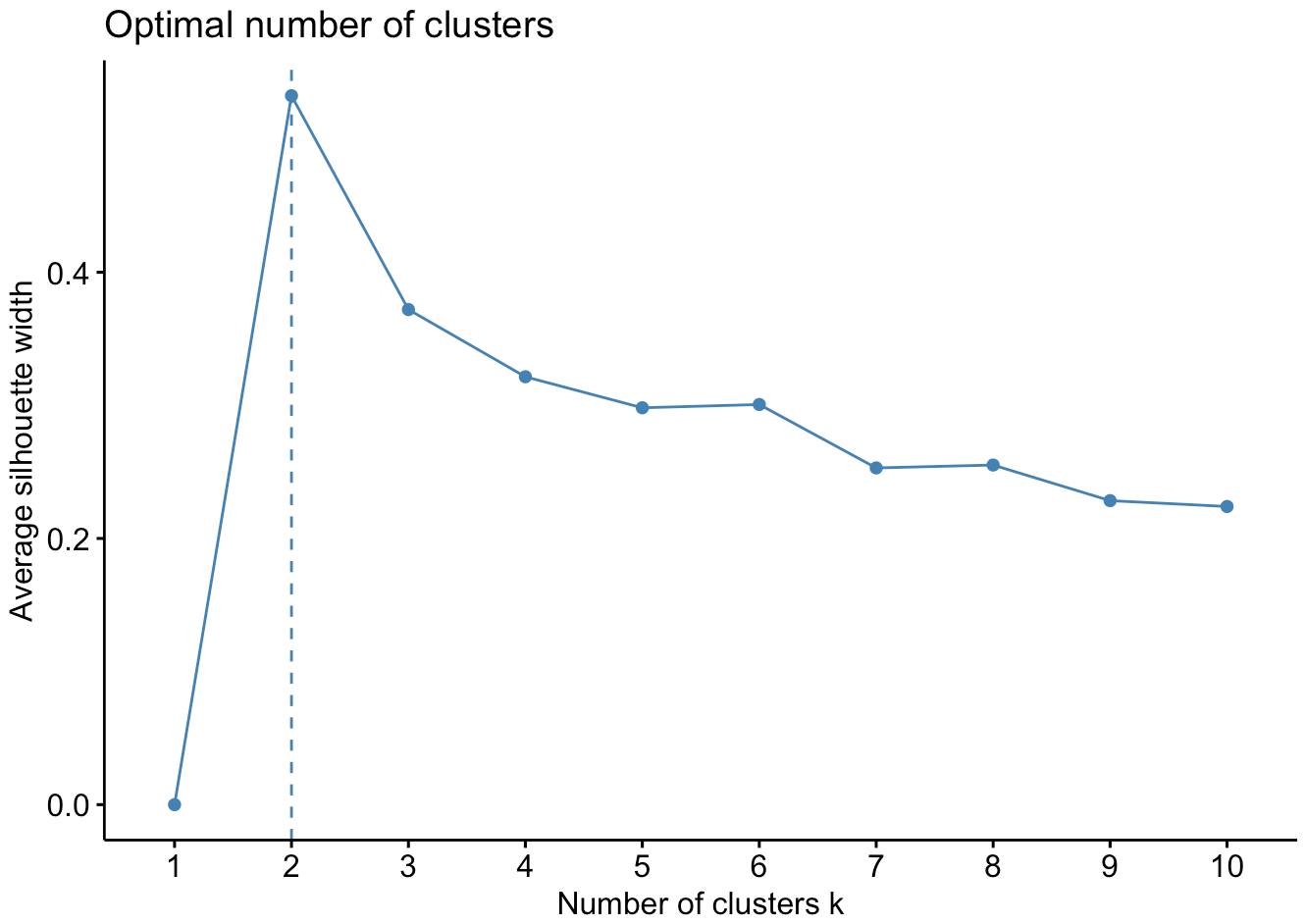
were to score high in PC2, it would mean that they have a moderately low urban population %, moderately low employment %, moderately high rural population %, moderately high percentage of people above the 65, moderate availability of sanitation facilities, moderate consumption of alcoholic beverages, moderately high % of COVID-19 confirmed cases and deaths, very high population, and very high amount of COVID-19 tests conducted yearly. We then visualized the PCs in a correlation circle and in terms of where the individual countries lay. Overall, the 2 PCs displayed on the graph explained about 58.6% of the total variation.

Clustering

```
# Prepare the data (drop the categorical variable 'Country' because it has too many cate
gories) for Gower dissimilarities
covid2020gow <- covid2020 %>%
  select(-Country)


# Apply 'gower' metric to the 'covid2020gow' dataset and save it as the matrix 'covid202
0_gower'
covid2020_gower <- daisy(covid2020gow, metric = "gower") %>%
  # Save as a matrix
  as.matrix


# Save an object 'test' which looks at the distances between pairs of countries
test <- covid2020_gower %>%
  # Save 'covid2020_gower' as a dataframe
  as.data.frame %>%
  # ID each row
  rownames_to_column("country1") %>%
  # Cross the ID of the country
  pivot_longer(-1, names_to = "country2", values_to = "distance") %>%
  # Get rid of pairs of the same country
  filter(country1 != country2) %>%
  # Avoid having the same pairs
  distinct(distance, .keep_all = TRUE)

#View the distances between the country pairs
 test
```

```
## # A tibble: 5,151 × 3
##    country1 country2 distance
##    <chr>    <chr>       <dbl>
##  1 1        2           0.221
##  2 1        3           0.170
##  3 1        4           0.111
##  4 1        5           0.178
##  5 1        6           0.121
##  6 1        7           0.282
##  7 1        8           0.142
##  8 1        9           0.251
##  9 1        10          0.135
## 10 1        11          0.176
## # … with 5,141 more rows
```

```
#Determine the number of optimum clusters to run for 'pam'
 fviz_nbclust(covid2020_gower, pam, method = "silhouette")
```




```
#Run pam clustering for 'covid2020_gower' with 2 clusters
 pam_results <- pam(covid2020_gower, k = 2, diss = TRUE)

#Have a look at the pam_results
pam_results
```

```
## Medoids:
##      ID
## [1,] "96" "96"
## [2,] "63" "63"
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
##   1   2   1   1   1   1   2   1   1   1   1   1   1   1   1   1   2   1   1   1
##  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
##   1   1   1   1   1   1   1   1   2   1   1   1   1   1   2   1   2   1   1   2
##  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
##   1   1   1   1   1   1   1   1   2   1   1   1   1   2   2   2   2   1   2   1
##  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
##   1   1   2   2   2   2   1   1   2   1   1   2   1   1   1   2   1   1   1   2
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100
##   1   2   1   1   1   1   2   1   1   1   2   2   1   1   2   1   1   1   1   1
## 101 102
##   2   2
## Objective function:
##     build      swap
## 0.1297519 0.1249905
##
## Available components:
## [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
## [6] "clusinfo"   "silinfo"    "diss"       "call"
```

```
#Determine the countries at the centers of each cluster
covid2020[96,]
```

```
## # A tibble: 1 × 12
##   Country `UrbanPop%` `unemploy%` `RuralPop%` `65above%` `Sanitation%`
##   <chr>         <dbl>       <dbl>       <dbl>      <dbl>         <dbl>
## 1 Ukraine        69.6        9.13        30.4       16.9          97.7
## # … with 6 more variables: `Alcohol%` <dbl>, Obesity <dbl>, `Confirmed%` <dbl>,
## #   `Death%` <dbl>, Population <dbl>, yearly_test <dbl>
```

```
covid2020[63,]
```

```
## # A tibble: 1 × 12
##   Country    `UrbanPop%` `unemploy%` `RuralPop%` `65above%` `Sanitation%`
##   <chr>            <dbl>       <dbl>       <dbl>      <dbl>         <dbl>
## 1 Mozambique        37.1        3.81        62.9       2.86          37.2
## # … with 6 more variables: `Alcohol%` <dbl>, Obesity <dbl>, `Confirmed%` <dbl>,
## #   `Death%` <dbl>, Population <dbl>, yearly_test <dbl>
```

```
#Determine the silhouette width for running pam with 2 clusters
pam_results$silinfo$avg.width
```

```
## [1] 0.4362948
```

```
#Now add the clustering results to the 'covid2020gow' dataset and overwrite that dataset
as 'covid_pam'
covid_pam <- covid2020gow %>%
  mutate(cluster = as.factor(pam_results$clustering)) %>% na.omit
```

First, we selected all variables besides the country variable and saved it as a new dataset called covid2020gow. We then calculated gower's distance between all the observations with the daisy function and saved it as a matrix named 'covid2020_gower'. Then with this dataset, using dplyr, we found the gower distance between each country and saved it as a new dataset called 'test'. Then, fviz_nbclust was used to find that the optimal number of clusters needed for our covid2020_gower dataset if pam clustering was run on it. Based on the fviz_nbclust, 2 clusters were done for pam because 2 clusters had the highest average silhouette width on this graph. With this, PAM was performed on our covid2020_gower dataset and saved to pam_results. Within pam_results, the center for cluster 1 was found to be Ukraine and the center of cluster 2 was found to be Mozambique. The silhouette width for running pam with 2 clusters was found to be about .436 which indcated that the structure was weak and could be artificial. The clustering results were then added to the covid2020gow dataset and saved to the covid_pam dataset.

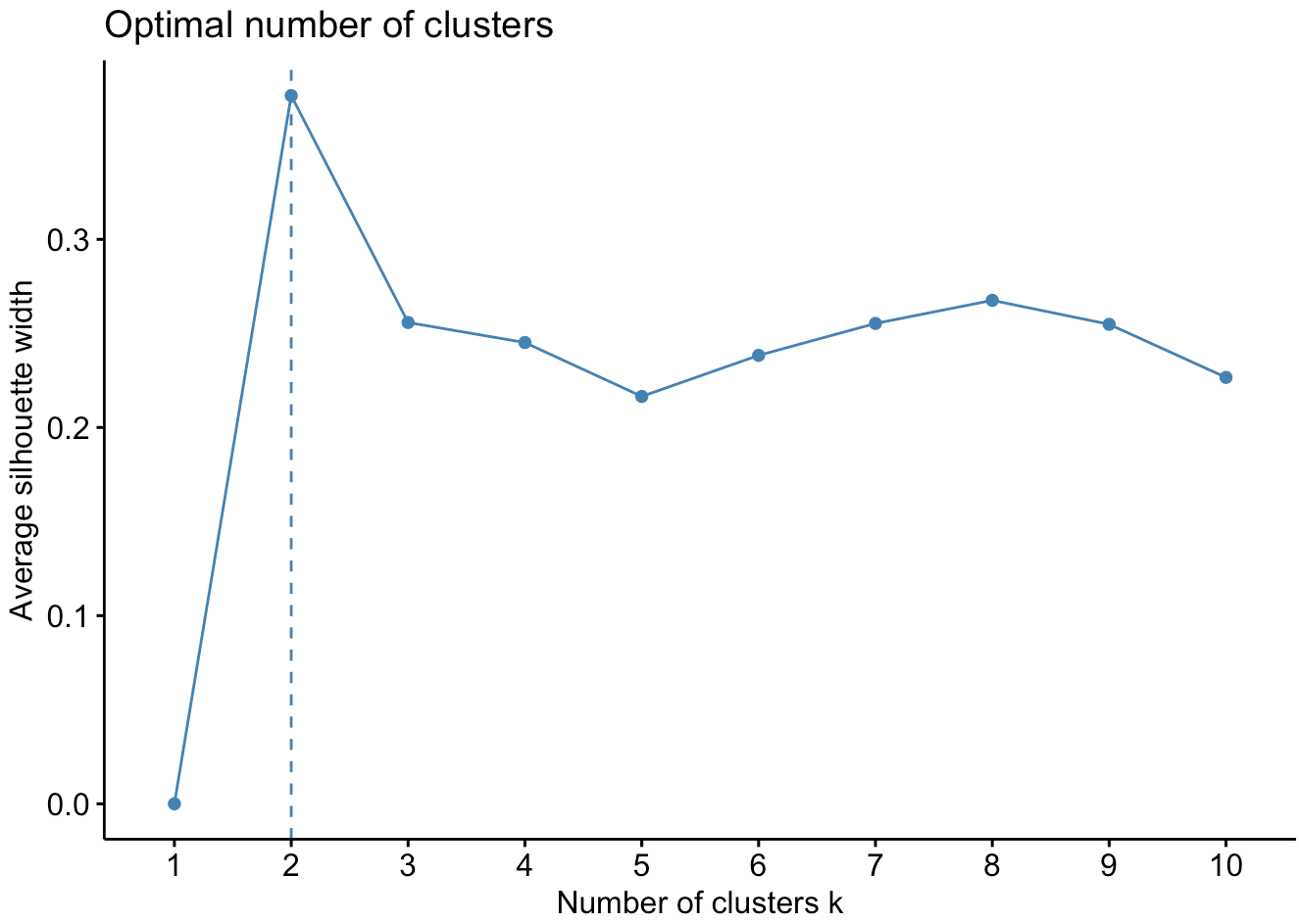
Pairwise Clustering Plot

```
# Visualize the clusters by showing all pairwise combinations of variables colored by cl
uster assignment
ggpairs(covid_pam, columns = 1:11, aes(color = cluster))
```

We visualized the clusters by showing all pairwise combinations of variables colored by clusters from the covid_pam dataset. The visualization shows the values for overall correlation and cluster-specific correlation between pairs of variables. The visualization also shows the distribution of values in the different clusters for 2 variables at a time in pairwise graphs.

##Clustering and PCA

```
# Import the HDI dataset for different countries around the world
HDI <- read_csv("HDI.csv")

# Inner join the covid2020 dataset with the HDI dataset and call the merged dataset 'cov
id2020hdi'
covid2020hdi <- covid2020 %>%
  inner_join(HDI, by = c("Country" = "country")) %>%
  #Remove the variable 'pop2022' from the merged dataset
  select(!pop2022) %>%
  # Add a variable named 'development' which has the value of 'developed' if HDI > 0.7 o
r the value of 'developing' if HDI < 0.7
  mutate(development = ifelse(hdi > .7, "developed", NA)) %>%
  mutate(development = ifelse(is.na(development) & hdi < .7, "developing", development))

# Remove NA values from the 'covid2020hdi' dataset
covid2020hdi <- covid2020hdi %>%
  na.omit()

# Determine the number of optimum clusters for pam on the 'covid2020scaled' dataset
fviz_nbclust(covid2020scaled, pam, method="silhouette")
```



```
# Run pam with 2 clusters for the 'covid2020scaled' dataset
pam_results2  <- covid2020scaled %>%
  pam(k = 2)

#Have a look at the pam_results2
pam_results2
```

```
## Medoids:
##    ID UrbanPop%  unemploy% RuralPop%   65above% Sanitation%   Alcohol%
## 96 91  0.212998  0.2345297 -0.212998  0.7433827   0.5828775 -0.0143861
## 63 59 -1.354486 -0.7760267  1.354486 -1.2682073  -1.9196184 -1.0260110
##      Obesity Confirmed%     Death%  Population yearly_test
## 96  0.618261  0.1490423  0.1048607 -0.09296293  -0.1377675
## 63 -1.600466 -1.0446309 -0.9795512 -0.14656258  -0.2700055
## Clustering vector:
##   1   4   5   6   7   8   9  10  12  13  14  15  16  17  18  19  20  21  22  23
##   1   1   1   1   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43
##   1   1   1   1   1   2   1   1   1   1   1   2   1   2   1   1   2   1   1   1
##  44  45  46  47  48  49  50  51  52  53  54  55  57  58  59  60  61  62  63  65
##   1   1   1   1   1   2   1   1   1   1   2   2   2   1   2   1   1   1   2   2
##  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
##   2   1   1   2   1   1   2   1   1   1   2   1   1   1   2   1   2   1   1   1
##  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102
##   1   2   1   1   1   1   2   1   1   2   1   1   1   1   1   2   2
## Objective function:
##    build     swap
## 2.440543 2.440543
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

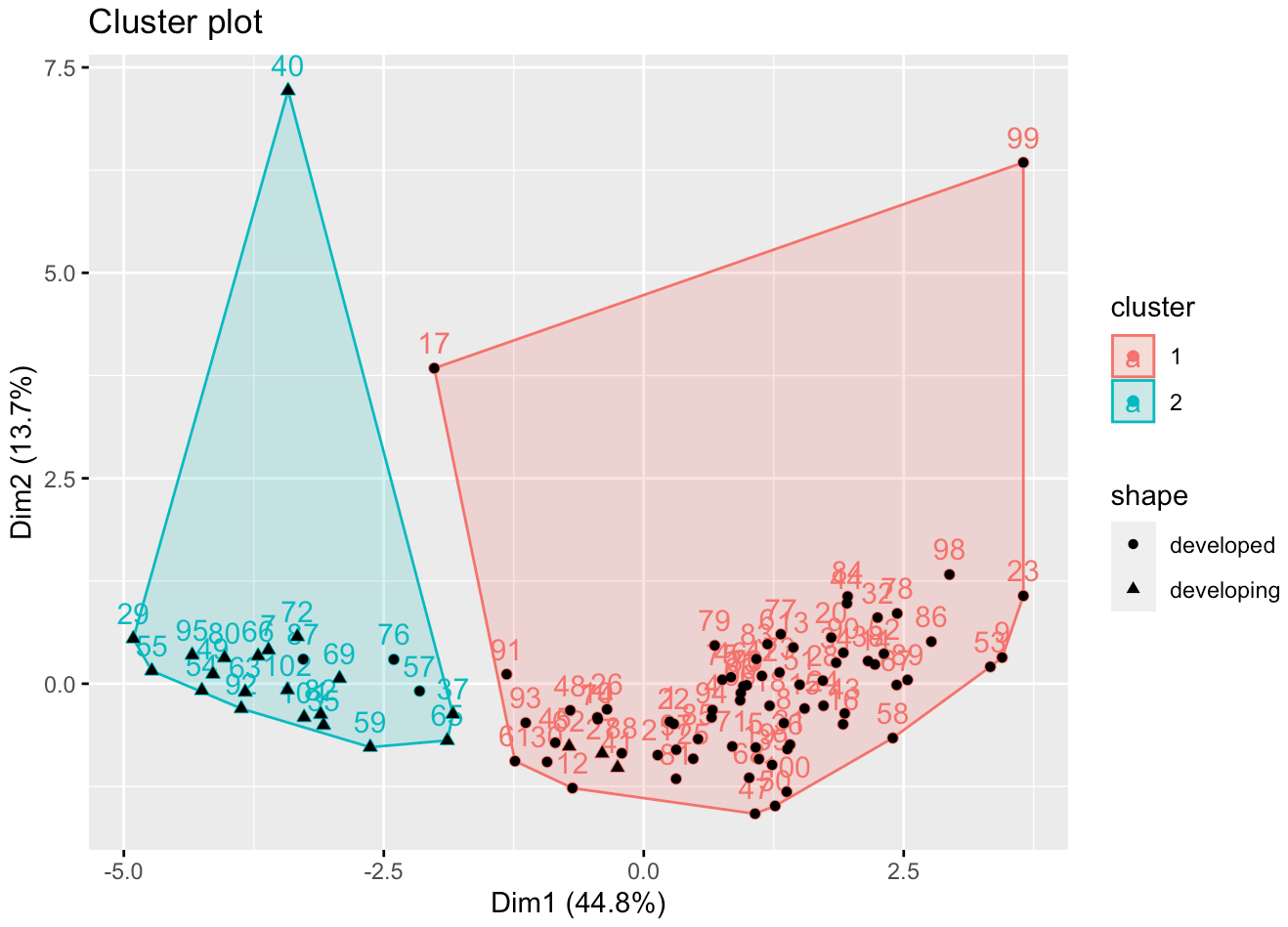
```
#Determine the countries at the centers of each cluster
covid2020[96,]
```

```
## # A tibble: 1 × 12
##   Country `UrbanPop%` `unemploy%` `RuralPop%` `65above%` `Sanitation%`
##   <chr>         <dbl>       <dbl>       <dbl>      <dbl>         <dbl>
## 1 Ukraine        69.6        9.13        30.4       16.9          97.7
## # … with 6 more variables: `Alcohol%` <dbl>, Obesity <dbl>, `Confirmed%` <dbl>,
## #   `Death%` <dbl>, Population <dbl>, yearly_test <dbl>
```

```
covid2020[63,]
```

```
## # A tibble: 1 × 12
##   Country    `UrbanPop%` `unemploy%` `RuralPop%` `65above%` `Sanitation%`
##   <chr>            <dbl>       <dbl>       <dbl>      <dbl>         <dbl>
## 1 Mozambique        37.1        3.81        62.9       2.86          37.2
## # … with 6 more variables: `Alcohol%` <dbl>, Obesity <dbl>, `Confirmed%` <dbl>,
## #   `Death%` <dbl>, Population <dbl>, yearly_test <dbl>
```

```
# Add the clustering results to the 'covid2020scaled' dataset
covidscaled_pam <- covid2020scaled %>%
  mutate(cluster = as.factor(pam_results2$clustering))



# Visualize the clustering results on a PCA graph making sure to show the development st
atus of the countries in each cluster
fviz_cluster(pam_results2, data = covid2020scaled,
             shape = as.factor(covid2020hdi$development)) +
  geom_point(aes(shape = as.factor(covid2020hdi$development))) +
  guides(shape = guide_legend(title = "shape"))
```




```
# Table showing the relation of clusters made to the development status of the countries
table(covidscaled_pam$cluster, covid2020hdi$development)
```

```
##
##     developed developing
##   1        71          3
##   2         3         20
```

```
#Calculate accuracy
(71+20)/97
```

```
## [1] 0.9381443
```

For this code chunk, we hoped to see how well our clusters matched the development status groups for the countries. To do this, we first imported the HDI dataset for different countries around the world. We then inner joined our 'covid2020' dataset with the HDI dataset by country names and kept all variables besides the pop2022 variable from the HDI dataset. We mutated the dataset by creating another variable called "development" in which countries with an hdi (human development index) score above .7 were classified as developed while countries with an hdi score below .7 were classified as developing. We then overwrote our 'covid2020hdi' dataset to exclude NA values. We utilized fvis_nbclust on our 'covid2020scaled' dataset from the PCA chunk of code above and found the optimal number of clusters needed for this scaled dataset was 2 as it had the highest average silhouette width. We then performed PAM on our 'covid2020scaled' dataset with a k value of 2 and the results were saved to pam_results2. Within pam_results2, the center for cluster 1 was found to be Ukraine and the center of cluster 2 was found to be Mozambique. We then mutated our 'covid2020' scaled dataset to include the clustering found from pam_results2 and called the variable cluster. We then visualized our clustering against our development variable using the fviz_cluster function in which the shapes represented the development variable while the color represented the cluster for each observation. We then used the table function to find the similarities between our development variable and clustering from our PAM and found that our clustering was about 93.8% accurate in determining the development status of each country.

##Classification

```
#For each country, assign the 'developed' status a value of '1' and 'developing' status
a value of 0
covid2020hdi <- covid2020hdi %>% mutate (actual = ifelse(hdi> 0.7, 1, 0))

# Add a new variable called 'positivityrate' which is a function of the variables 'Confi
rmed%', 'Population', 'yearly_test'
covid2020hdi <- covid2020hdi %>% mutate(positivityrate = (( `Confirmed%`/100)*Population
*100)/(yearly_test))

# Use a glm model to create a fit which shows development status based on 'positivityrat
e', 'Confirmed%', and 'Death%'
fit <- glm(actual ~ positivityrate + `Confirmed%` + `Death%`, data = covid2020hdi, famil
y = "binomial")
summary(fit)
```

```
##
## Call:
## glm(formula = actual ~ positivityrate + `Confirmed%` + `Death%`,
##     family = "binomial", data = covid2020hdi)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.91122   0.00326   0.08566   0.42729   1.53137
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.8032     0.4303  -1.867   0.0620 .
## positivityrate  -0.2896     0.8479  -0.342   0.7327
## `Confirmed%`     1.4288     0.5893   2.424   0.0153 *
## `Death%`         7.3647    20.5058   0.359   0.7195
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 106.260  on 96  degrees of freedom
## Residual deviance:  62.763  on 93  degrees of freedom
## AIC: 70.763
##
## Number of Fisher Scoring iterations: 7
```

```
# Calculate a predicted probability based on the fit
log_covid2020hdi <- covid2020hdi %>%
  mutate(score = predict(fit, type = "response"),
         predicted = ifelse(score < 0.5, 0, 1))
log_covid2020hdi
```

```
## # A tibble: 97 × 18
##    Country          `UrbanPop%` `unemploy%` `RuralPop%` `65above%` `Sanitation%`
##    <chr>                  <dbl>       <dbl>       <dbl>      <dbl>         <dbl>
##  1 Albania                 62.1       13.3         37.9       14.7          99.3
##  2 Armenia                 63.3       21.2         36.7       11.8          93.9
##  3 Australia               86.2        6.46        13.8       16.2         100.
##  4 Austria                 58.7        5.36        41.3       19.2         100.
##  5 Bangladesh              38.2        5.41        61.8        5.23         54.2
##  6 Belarus                 79.5        4.77        20.5       15.6          97.9
##  7 Belgium                 98.1        5.55         1.92      19.3          99.5
##  8 Bolivia                 70.1        7.90        29.9        7.49         65.8
##  9 Botswana                70.9       24.9         29.1        4.51         80.0
## 10 Brazil                  87.1       13.7         12.9        9.59         90.1
## 11 Bulgaria                75.7        5.12        24.3       21.5          86.0
## 12 Canada                  81.6        9.46        18.4       18.1          99.0
## 13 Chile                   87.7       11.2         12.3       12.2         100.
## 14 China                   61.4        5           38.6       12.0          92.4
## 15 Colombia                81.4       15.0         18.6        9.06         93.7
## 16 Costa Rica              80.8       17.4         19.2       10.3          97.9
## 17 Croatia                 57.6        7.51        42.4       21.3          96.6
## 18 Cuba                    77.2        2.62        22.8       15.9          91.4
## 19 Cyprus                  66.8        7.59        33.2       14.4          99.4
## 20 Czech Republic          74.1        2.55        25.9       20.1          99.1
## 21 Denmark                 88.1        5.64        11.9       20.2          99.6
## 22 Dominican Repub…        82.5        6.13        17.5        7.53         87.2
## 23 Ecuador                 64.2        6.11        35.8        7.59         91.5
## 24 El Salvador             73.4        6.25        26.6        8.65         82.4
## 25 Estonia                 69.2        6.80        30.8       20.4          99.1
## 26 Ethiopia                21.7        3.24        78.3        3.54          8.91
## 27 Fiji                    57.2        4.72        42.8        5.82         99.2
## 28 Finland                 85.5        7.76        14.5       22.6          99.4
## 29 France                  81.0        8.01        19.0       20.8          98.6
## 30 Georgia                 59.5       18.5         40.5       15.3          85.8
## 31 Germany                 77.5        3.81        22.5       21.7          99.2
## 32 Ghana                   57.3        4.65        42.7        3.14         23.7
## 33 Greece                  79.7       16.3         20.3       22.3          99.0
## 34 Guatemala               51.8        3.55        48.2        5.04         67.9
## 35 Hungary                 71.9        4.25        28.1       20.2          98.0
## 36 Iceland                 93.9        5.48         6.10      15.6          98.8
## 37 India                   34.9        8.00        65.1        6.57         71.3
## 38 Iraq                    70.9       14.1         29.1        3.44        100.
## 39 Ireland                 63.7        5.62        36.3       14.6          91.3
## 40 Israel                  92.6        4.33         7.41      12.4          99.9
## 41 Italy                   71.0        9.16        29.0       23.3          99.9
## 42 Jamaica                 56.3        9.48        43.7        9.08         86.6
## 43 Japan                   91.8        2.80         8.22      28.4          99.9
## 44 Jordan                  91.4       19.0          8.58       3.95         97.1
## 45 Kazakhstan              57.7        4.89        42.3        7.90         97.9
## 46 Kenya                   28.0        5.73        72.0        2.51         32.7
## 47 Kuwait                 100          3.54         0          3.04        100
## 48 Latvia                  68.3        8.10        31.7       20.7          92.4
## 49 Lithuania               68.0        8.49        32.0       20.6          93.9
```

```
## 50 Luxembourg              91.5        6.77        8.55      14.4          97.6
## 51 Madagascar              38.5        2.47       61.5        3.10         12.3
## 52 Malawi                  17.4        6.70       82.6        2.64         26.6
## 53 Maldives                40.7        6.33       59.3        3.59         99.2
## 54 Malta                   94.7        4.26        5.26      21.3         100.
## 55 Mauritania              55.3       11.3        44.7        3.18         49.8
## 56 Mexico                  80.7        4.45       19.3        7.62         92.4
## 57 Mongolia                68.7        7.01       31.3        4.31         67.7
## 58 Morocco                 63.5       11.5        36.5        7.61         87.3
## 59 Mozambique              37.1        3.81       62.9        2.86         37.2
## 60 Namibia                 52.0       21.4        48.0        3.59         35.3
## 61 Nepal                   20.6        4.72       79.4        5.83         76.6
## 62 Netherlands             92.2        3.82        7.76      20.0          97.7
## 63 New Zealand             86.7        4.59       13.3       16.4         100.
## 64 Nigeria                 52.0        9.71       48.0        2.74         42.7
## 65 North Macedonia         58.5       17.2        41.5       14.5          98.3
## 66 Norway                  83.0        4.42       17.0       17.5          98.1
## 67 Pakistan                37.2        4.30       62.8        4.35         68.4
## 68 Panama                  68.4       12.9        31.6        8.54         84.6
## 69 Paraguay                62.2        7.55       37.8        6.81         92.7
## 70 Peru                    78.3        7.18       21.7        8.73         78.6
## 71 Philippines             47.4        2.52       52.6        5.51         82.3
## 72 Poland                  60.0        3.16       40.0       18.7         100.
## 73 Portugal                66.3        6.79       33.7       22.8          99.6
## 74 Romania                 54.2        5.03       45.8       19.2          87.1
## 75 Rwanda                  17.4        1.49       82.6        3.12         68.8
## 76 Saudi Arabia            84.3        7.45       15.7        3.50        100
## 77 Senegal                 48.1        3.62       51.9        3.11         56.8
## 78 Serbia                  56.4        9.01       43.6       19.1          97.9
## 79 Slovenia                55.1        4.97       44.9       20.7          98.1
## 80 South Africa            67.4       29.2        32.6        5.51         78.5
## 81 Spain                   80.8       15.5        19.2       20.0          99.9
## 82 Sri Lanka               18.7        5.88       81.3       11.2          93.7
## 83 Suriname                66.1        9.78       33.9        7.13         90.0
## 84 Sweden                  88.0        8.29       12.0       20.3          99.3
## 85 Switzerland             73.9        4.82       26.1       19.1          99.9
## 86 Thailand                51.4        1.10       48.6       13.0          98.7
## 87 Togo                    42.8        3.94       57.2        2.91         18.6
## 88 Trinidad and To…        53.2        4.57       46.8       11.5          93.9
## 89 Turkey                  76.1       13.1        23.9        8.98         99.2
## 90 Uganda                  25.0        2.77       75.0        1.99         19.8
## 91 Ukraine                 69.6        9.13       30.4       16.9          97.7
## 92 United Arab Emi…        87.0        3.19       13.0        1.26         99.2
## 93 United Kingdom          83.9        4.47       16.1       18.7          99.1
## 94 United States           82.7        8.05       17.3       16.6          99.7
## 95 Uruguay                 95.5       10.4         4.49      15.1          98.1
## 96 Zambia                  44.6       12.8        55.4        2.13         31.9
## 97 Zimbabwe                32.2        5.35       67.8        3.01         35.2
## # … with 12 more variables: `Alcohol%` <dbl>, Obesity <dbl>,
## #   `Confirmed%` <dbl>, `Death%` <dbl>, Population <dbl>, yearly_test <dbl>,
## #   hdi <dbl>, development <chr>, actual <dbl>, positivityrate <dbl>,
## #   score <dbl>, predicted <dbl>
```

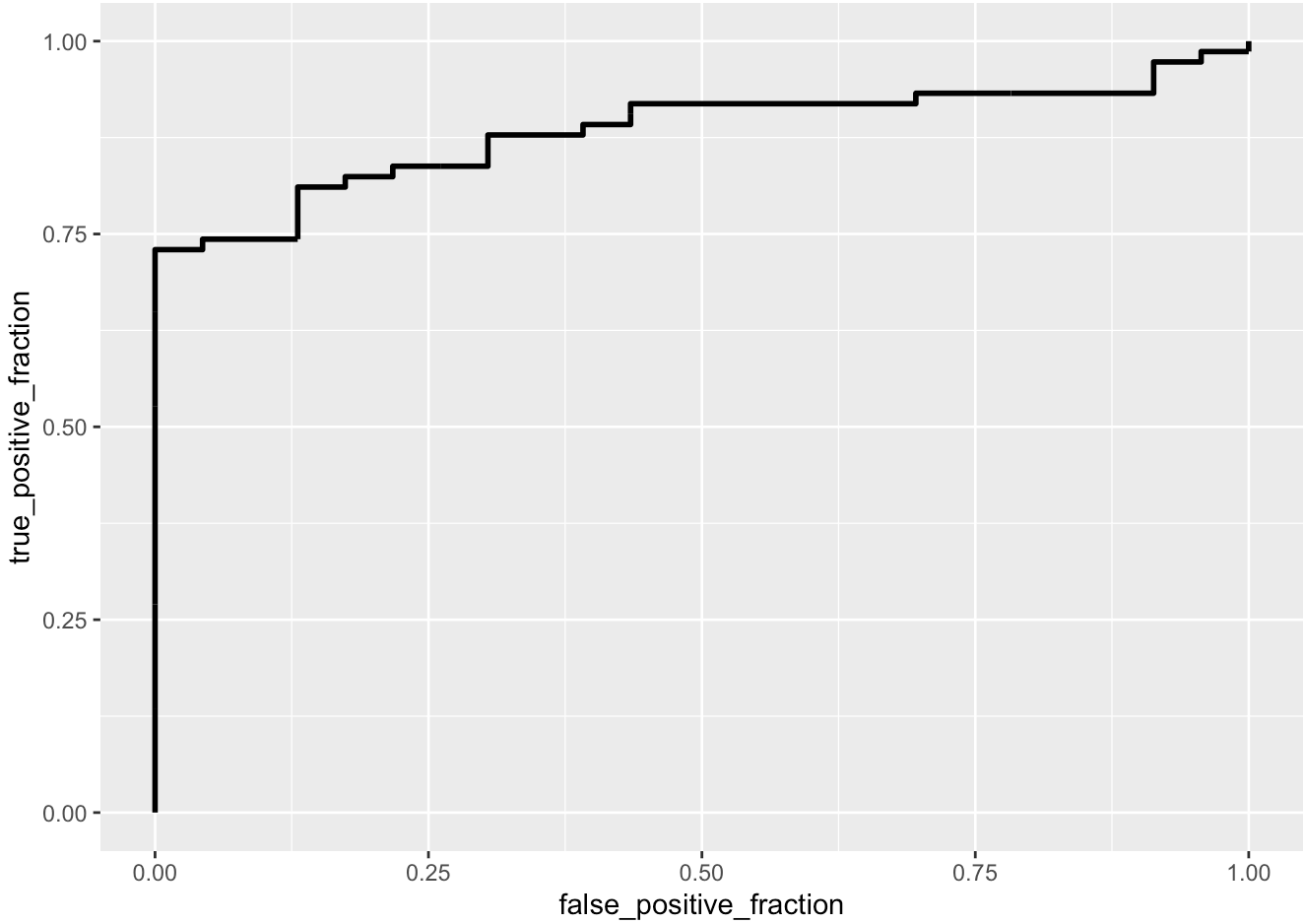
```
# Confusion matrix: compare true to predicted condition
table(log_covid2020hdi$actual, log_covid2020hdi$predicted) %>% addmargins
```

```
##
##        0  1 Sum
##   0   16  7  23
##   1   10 64  74
##   Sum 26 71  97
```

```
# Calculate accuracy based on confusion matric
80/97
```

```
## [1] 0.8247423
```

```
# Visualize a ROC curve for the glm model
ROC <- ggplot(log_covid2020hdi) +
  geom_roc(aes(d = actual, m = score), n.cuts = 0)
ROC
```




```
# Calculate the area under the curve for the ROC model
calc_auc(ROC)
```

```
##   PANEL group       AUC
## 1     1    -1 0.8830787
```

The covid2020hdi dataset was mutated by adding a new variable called "actual" in which countries with an hdi scores above .7, or developed countries, were given a value of 1 while countries with an hdi score below .7 (developing countries) were given a value of 0. Also, a new variable called positivityrate was added to the dataset in which positivityrate was a function of confirmed%, population, and yearly_test. We then used the glm function on our covid2020hdi data to create a fit which demonstrated the development status based on positivityrate, confirmed%, and death%. This fit was then summarized and displayed. We then calculated a predicted probability based on the fit and saved it to log_covid2020hdi. We then used the table function to compare the predicted and true values. The accuracy was found to be about 82.5%. We then visualized a ROC curve for the glm model and saved it to ROC and found the area under the curve of the ROC model to be about 0.883 which indicated that the model was a good fit.

##Cross-validation with the 'k-fold' method

```
# Choose number of folds
k = 10

# Randomly order rows in the dataset
data <- covid2020hdi[sample(nrow(covid2020hdi)), ]

#Set Seed to save results
set.seed(11)

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

# Use a for loop to get diagnostics for each test set
diags_k <- NULL

for(i in 1:k){
  # Create training and test sets
  train <- data[folds != i, ] # all observations except in fold i
  test <- data[folds == i, ]  # observations in fold i

  # Train model on training set (all but fold i)
  fit <- glm(actual ~ positivityrate + `Confirmed%` + `Death%`, data = train, family =
"binomial")

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

  # Consider the ROC curve for the test dataset
  ROC <- ggplot(df) +
    geom_roc(aes(d = actual, m = probability, n.cuts = 0))

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

# Average performance
mean(diags_k)
```

```
## [1] 0.8731151
```

The k-fold cross-validation method was used to see if there were any signs of overfitting. The code chunk above cut the data into 10 folds in which 9 folds were used as the training set and the other 1 fold was used as the test set, this process was repeated till each fold was at least used once as the test set. The average performance obtained from the k-folds method indicated an average area the curve of 0.881 which is very close to the area under the curve of the original model which was 0.883. Thus, the model for the entire dataset does not show any signs of over-fitting as the average AUC of the k-folds cross-validation method almost matched the AUC of the original model, indicating that the model would remain a good fit even if any new data is added.

References

'Health Nutrition and Population Statistics' (HealthStats.csv): Website Link (https://databank.worldbank.org/source/health-nutrition-and-population-statistics)

Context: This dataset was obtained from World Bank, and this dataset contained many population and health statistics for countries around the world.

'diet2020' (Food_Supply_kcal_Data.csv): Website Link (https://www.kaggle.com/datasets/mariaren/covid19-healthy-diet-dataset)

Context: This dataset was obtained from Kaggle, and this dataset contained many dietary intake variables, health conditions associated with nutrition, and COVID-19 related data for countries around the world.

'totaltests' (total-tests-for-covid-19.csv): Website Link (https://ourworldindata.org/grapher/full-list-total-tests-for-covid-19?time=latest)

Context: This dataset was obtained from the Our World in Data website and contained the total COVID-19 tests conducted daily in many countries around the world

'HDI' (HDI.csv): Website Link (https://worldpopulationreview.com/country-rankings/hdi-by-country)

Context: The dataset shows the HDI values and population values of different countries around the world