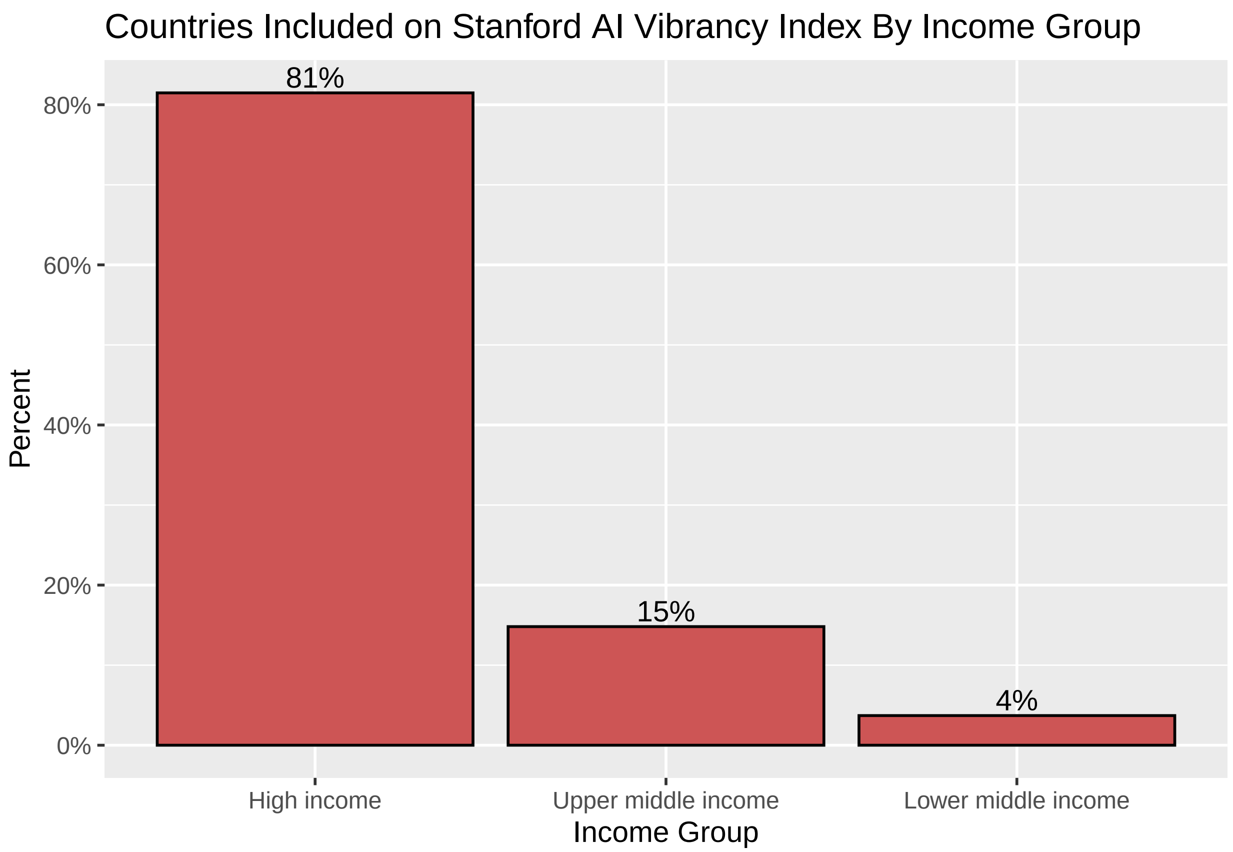
AI Vibrancy EDA

Taylor Boeckman

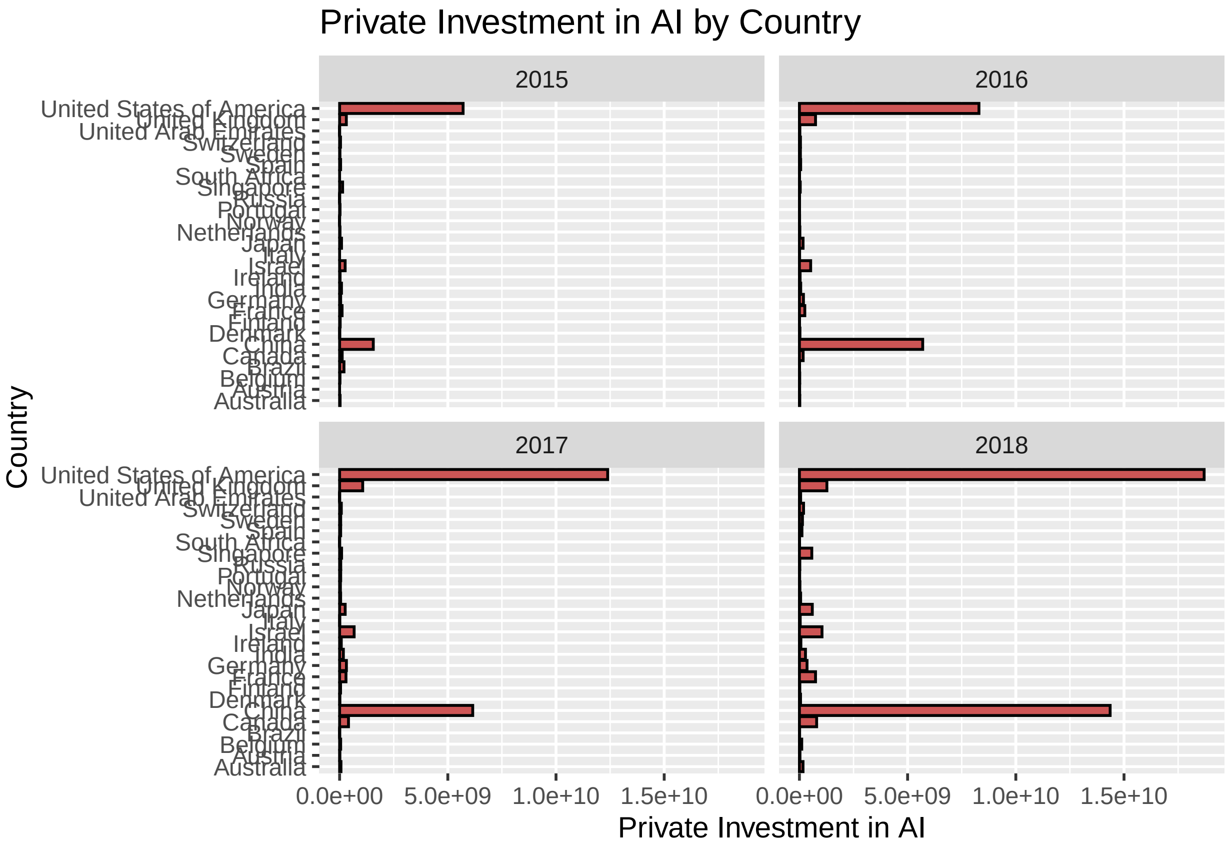
4/22/2020

```r
aiVibrancyIndicators2019 = read.csv("2019indicators.csv")

aiVibrancyIndicators2019.df <- aiVibrancyIndicators2019

aiVibrancyIndicators2019.tib <- as_tibble(aiVibrancyIndicators2019.df) %>%
  mutate( female_ai_authors = as.double( female_ai_authors ) ) %>%
  mutate( female_ai_skill_penetration = as.double( female_ai_skill_penetration ) )
```

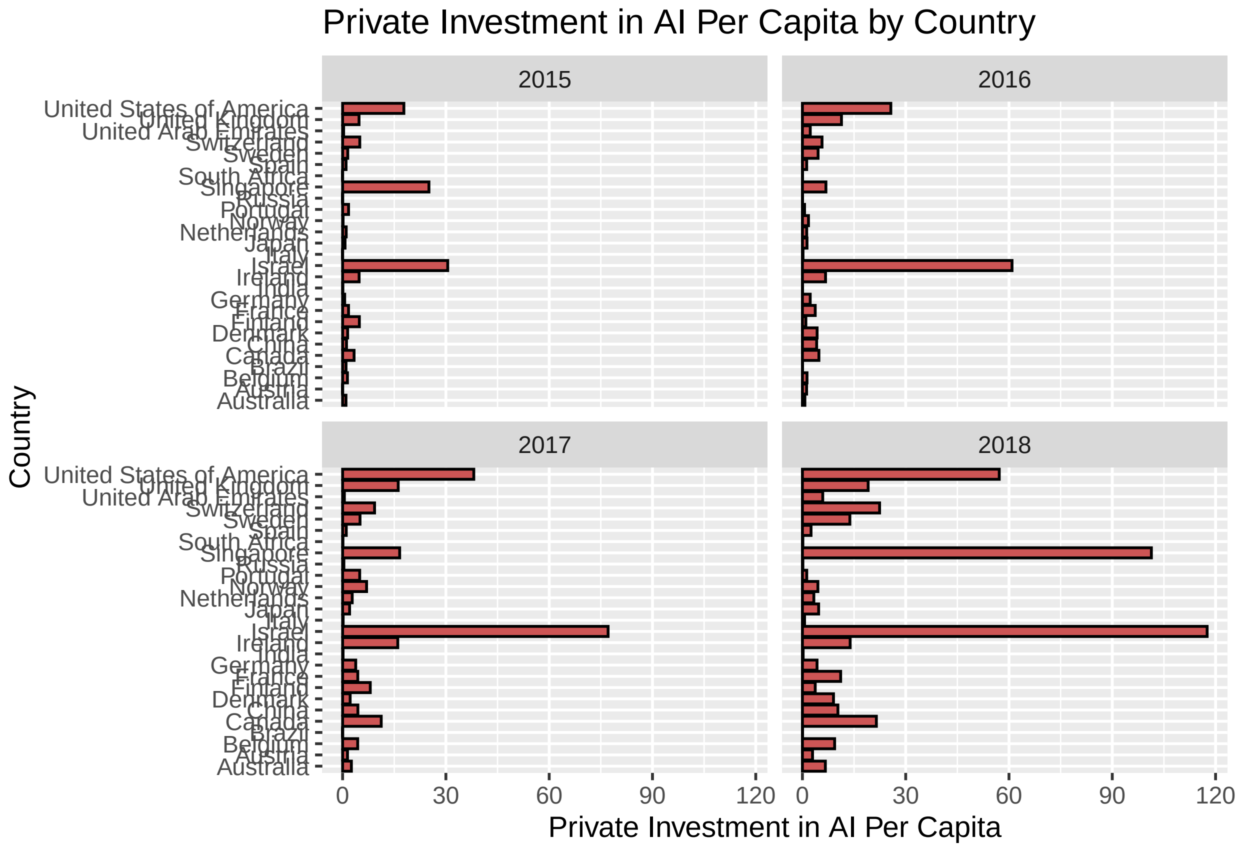
```r
aiVibrancyIndicators2019.tib %>%
  count( income.group ) %>%
  mutate( pct = n / sum(n),
          pctlabel = paste0( round( pct*100 ), "%" ) ) %>%
  ggplot( aes( x = reorder( income.group, -pct ),
               y = pct ) ) +
      geom_col( fill = "indianred3",
                color = "black" ) +
      geom_text( aes( label = pctlabel ),
                      vjust = -0.25 ) +
      scale_y_continuous( labels = scales::percent ) +
      labs( x = "Income Group",
            y = "Percent",
            title = "Countries Included on Stanford AI Vibrancy Index By Income Group" )
```



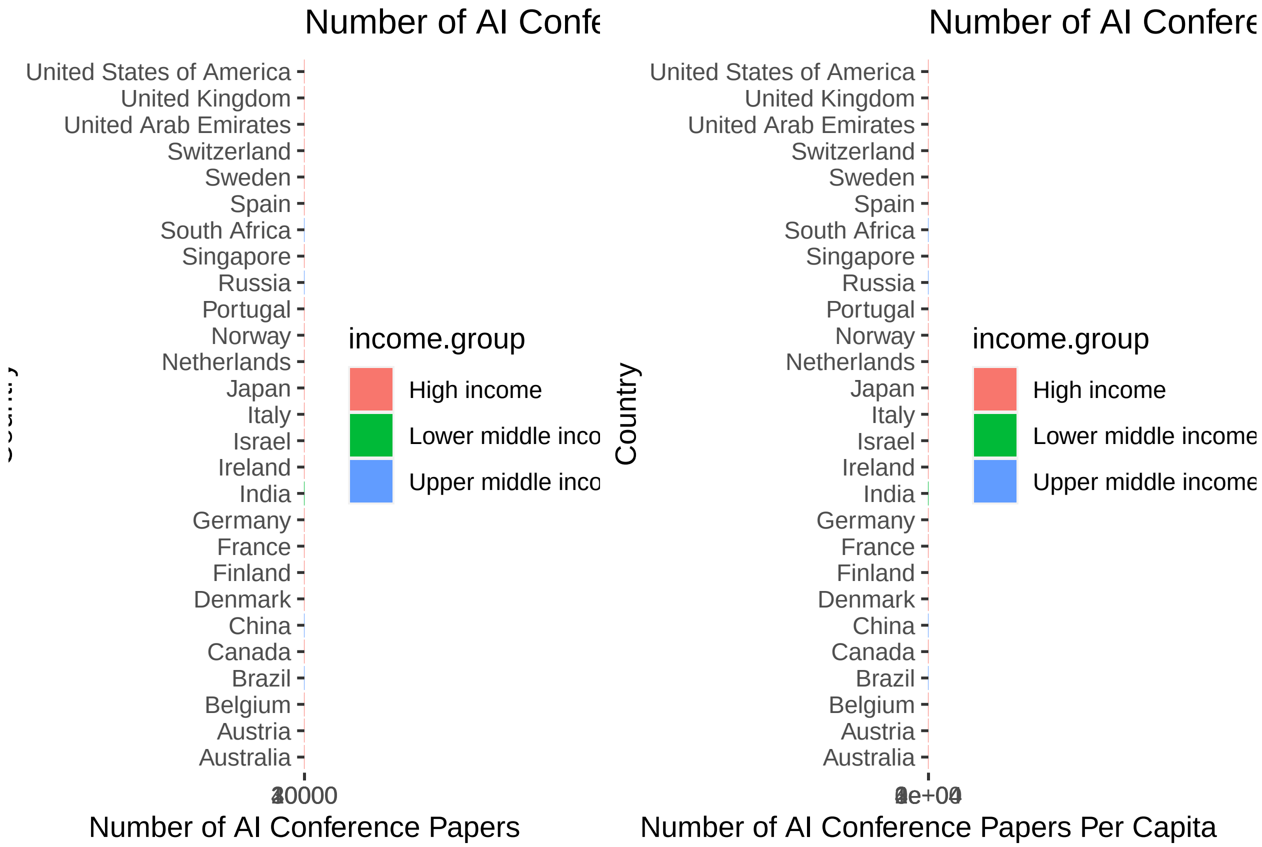
```
aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = ai_privateinvestment_tot , y = country ) ) +
      geom_col( fill = "indianred3",
                color = "black" ) +
      facet_wrap( ~ year, ncol = 2) +
      labs( x = "Private Investment in AI",
            y = "Country",
            title = "Private Investment in AI by Country" )
```



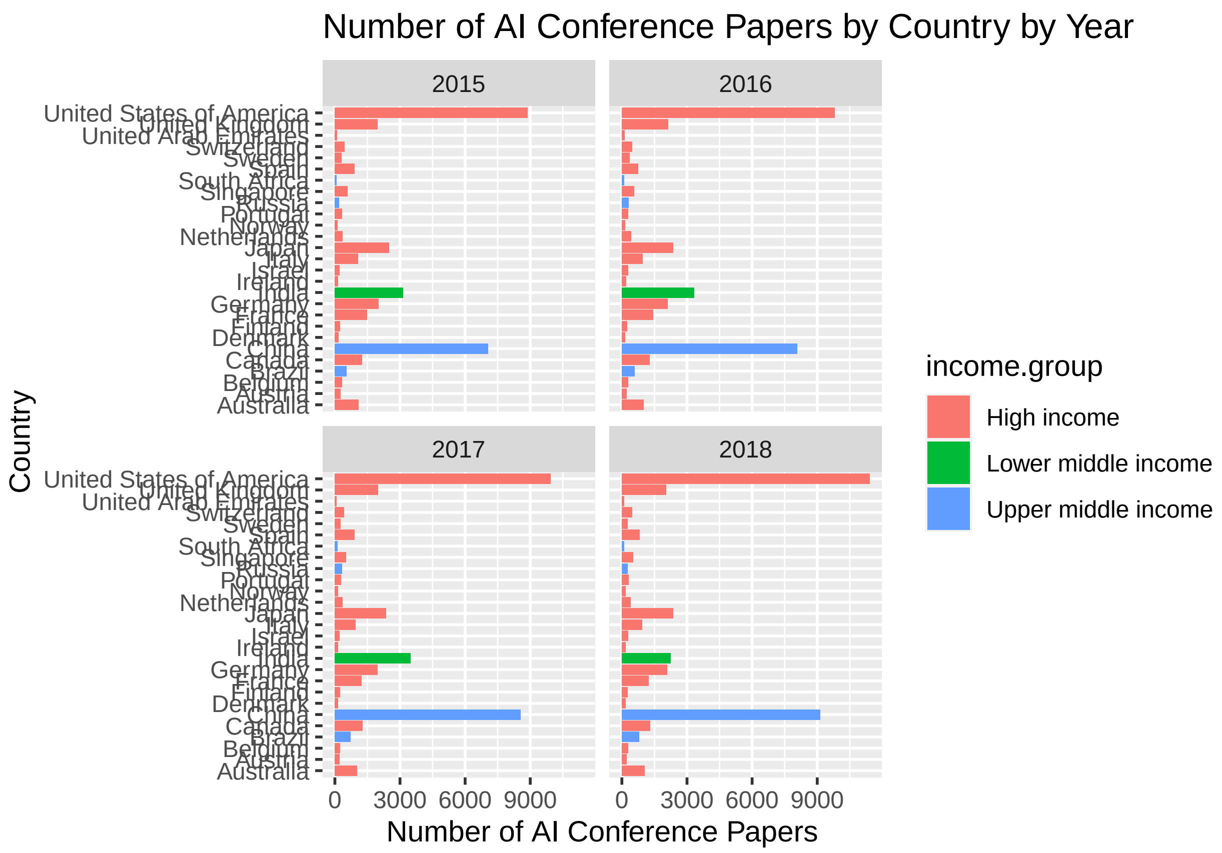
```
aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = ai_privateinvestment_tot_pc , y = country ) ) +
      geom_col( fill = "indianred3",
                color = "black" ) +
      facet_wrap( ~ year, ncol = 2) +
      labs( x = "Private Investment in AI Per Capita",
            y = "Country",
            title = "Private Investment in AI Per Capita by Country" )
```



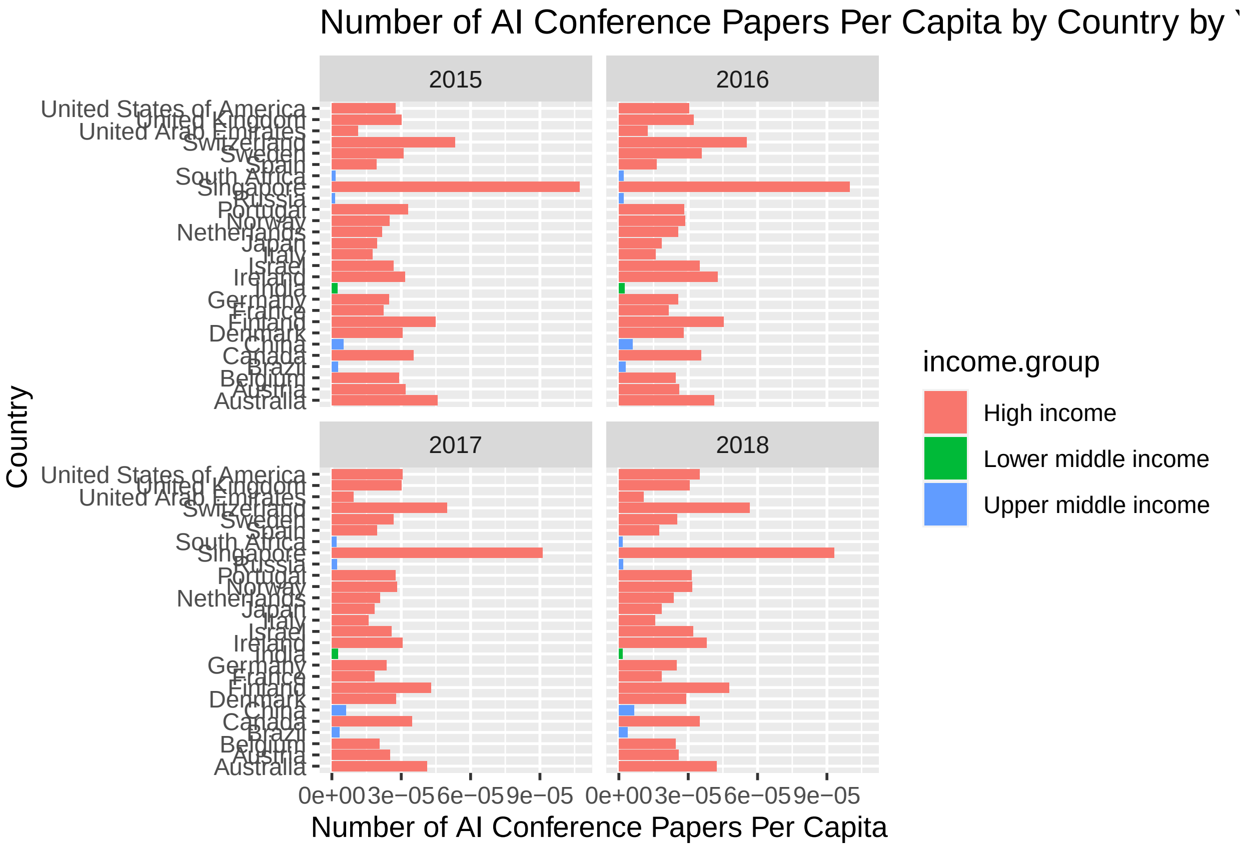
```
num_AIconf_papers_graph <- aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = num_AIconf_papers, y = country, fill = income.group ) ) +
      geom_col() +
      labs( x = "Number of AI Conference Papers",
            y = "Country",
            title = "Number of AI Conference Papers by Country" )

num_AIconf_papers_pc_graph <- aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = num_AIconf_papers_pc, y = country, fill = income.group ) ) +
      geom_col() +
      labs( x = "Number of AI Conference Papers Per Capita",
            y = "Country",
            title = "Number of AI Conference Papers  Per Capita by Country" )

grid.arrange(num_AIconf_papers_graph, num_AIconf_papers_pc_graph, ncol = 2)
```



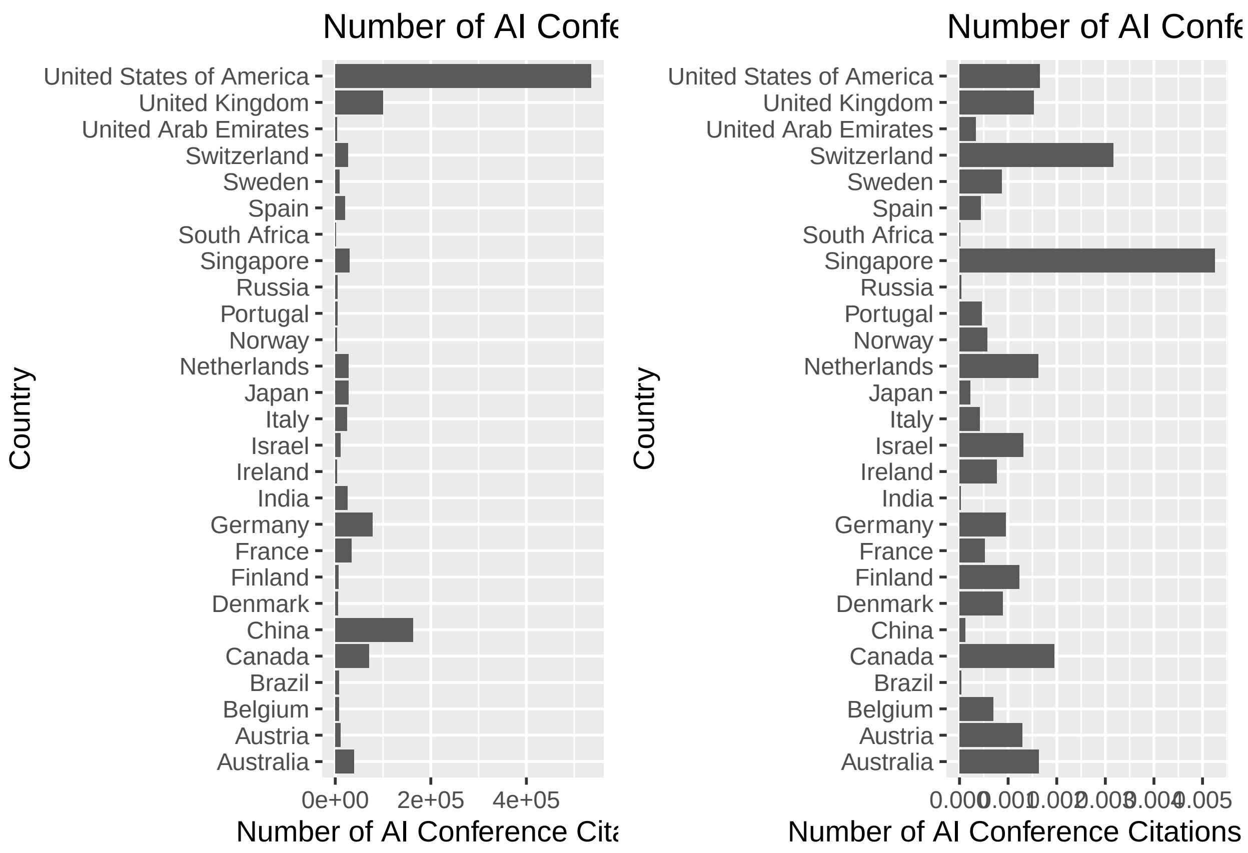
```
aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = num_AIconf_papers, y = country, fill = income.group ) ) +
      geom_col() +
      facet_wrap( ~ year, ncol = 2 ) +
      labs( x = "Number of AI Conference Papers",
            y = "Country",
            title = "Number of AI Conference Papers by Country by Year" )
```



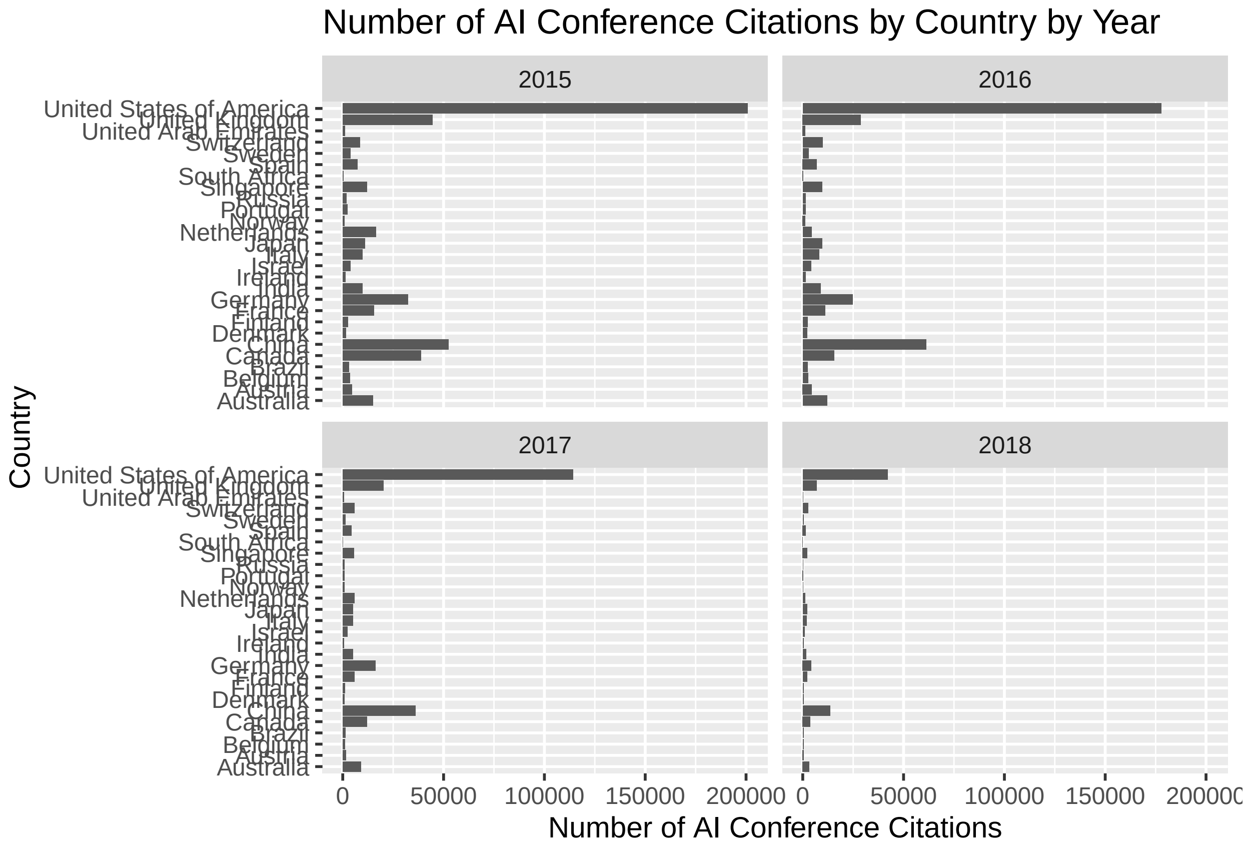
```r
aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = num_AIconf_papers_pc, y = country, fill = income.group ) ) +
      geom_col() +
      facet_wrap( ~ year, ncol = 2 ) +
      labs( x = "Number of AI Conference Papers Per Capita",
            y = "Country",
            title = "Number of AI Conference Papers Per Capita by Country by Year" )
```



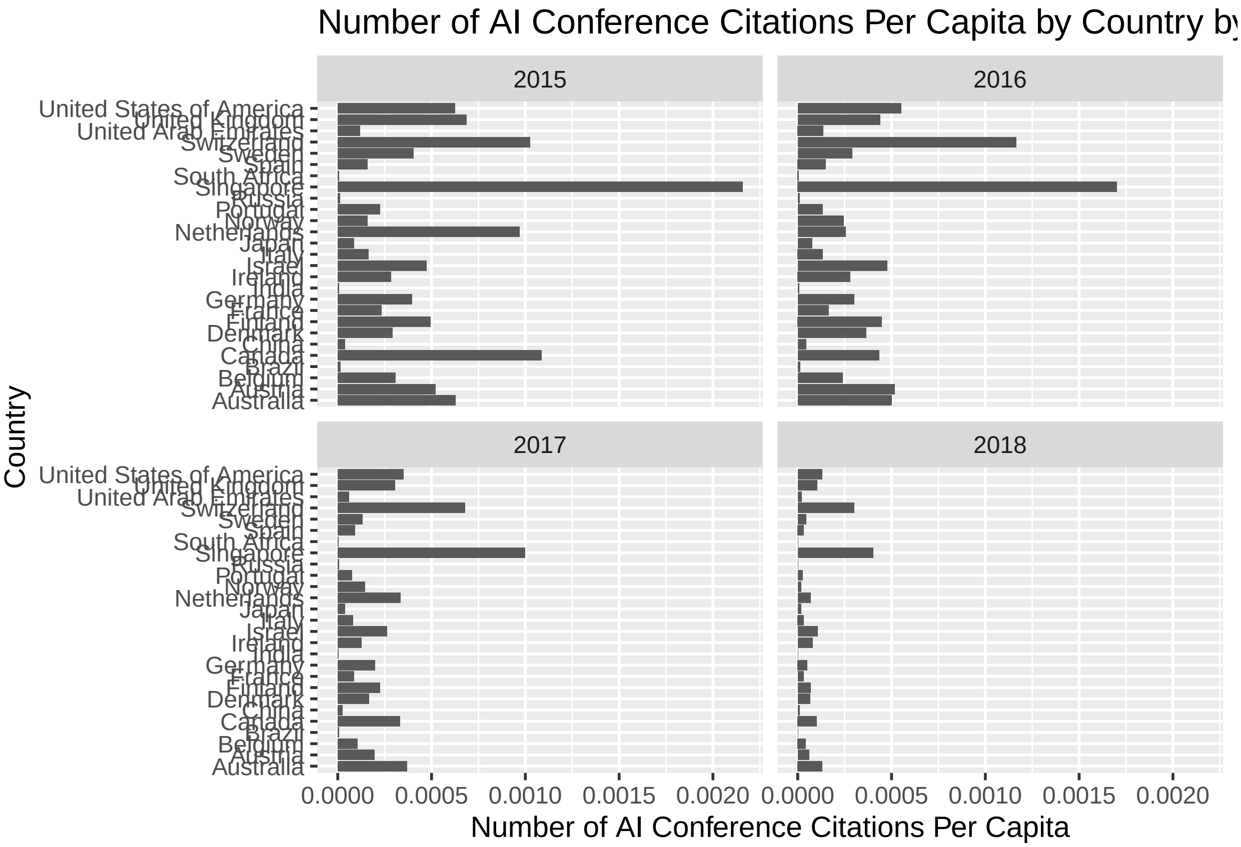
```
num_AIconf_citation_graph <- aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = num_AIconf_citation, y = country ) ) +
      geom_col() +
      labs( x = "Number of AI Conference Citations",
            y = "Country",
            title = "Number of AI Conference Citations by Country" )


num_AIconf_citation_pc_graph <- aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = num_AIconf_citation_pc, y = country ) ) +
      geom_col() +
      labs( x = "Number of AI Conference Citations Per Capita",
            y = "Country",
            title = "Number of AI Conference Citations Per Capita by Country" )

grid.arrange(num_AIconf_citation_graph, num_AIconf_citation_pc_graph, ncol = 2)
```



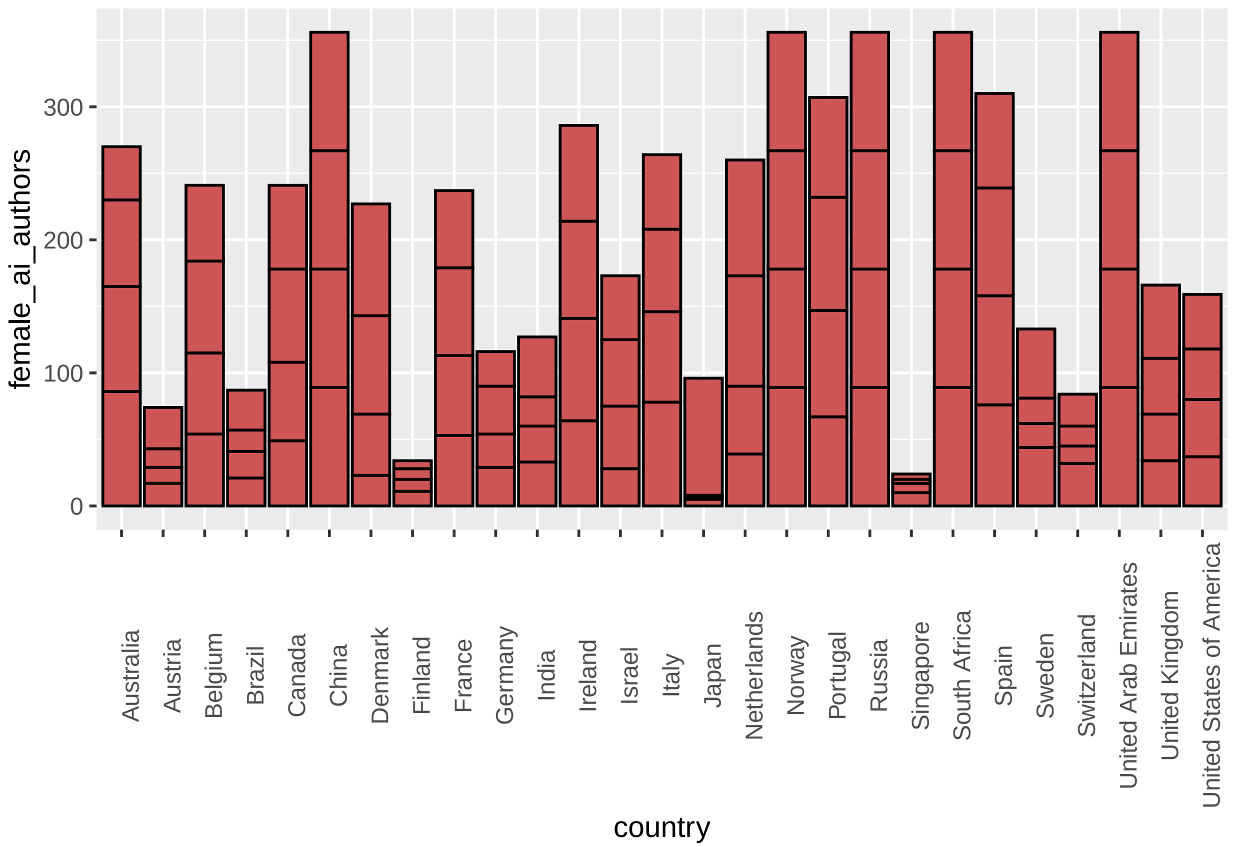
```
aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = num_AIconf_citation, y = country ) ) +
      geom_col() +
      facet_wrap( ~ year, ncol = 2 ) +
      labs( x = "Number of AI Conference Citations",
            y = "Country",
            title = "Number of AI Conference Citations by Country by Year" )
```



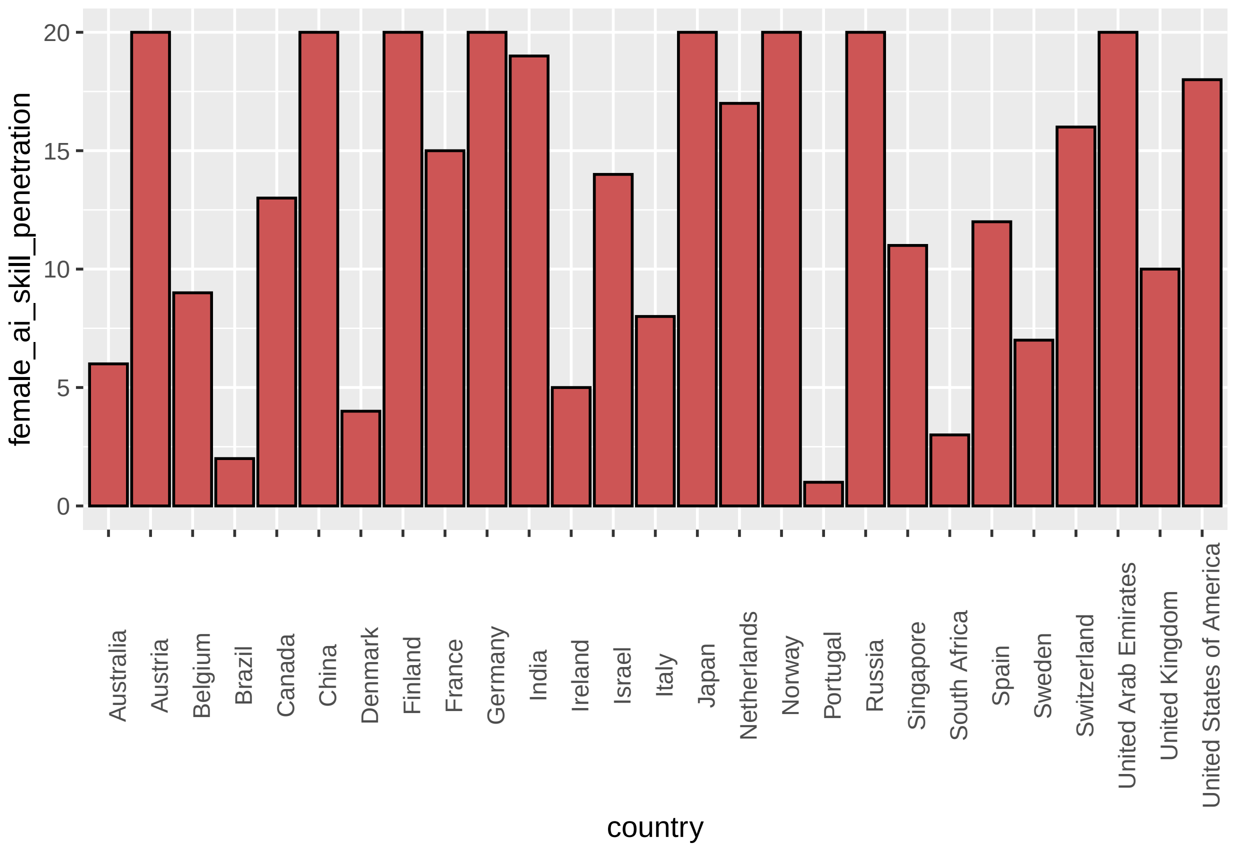
```r
aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = num_AIconf_citation_pc, y = country ) ) +
      geom_col() +
      facet_wrap( ~ year, ncol = 2 ) +
      labs( x = "Number of AI Conference Citations Per Capita",
            y = "Country",
            title = "Number of AI Conference Citations Per Capita by Country by Year" )
```



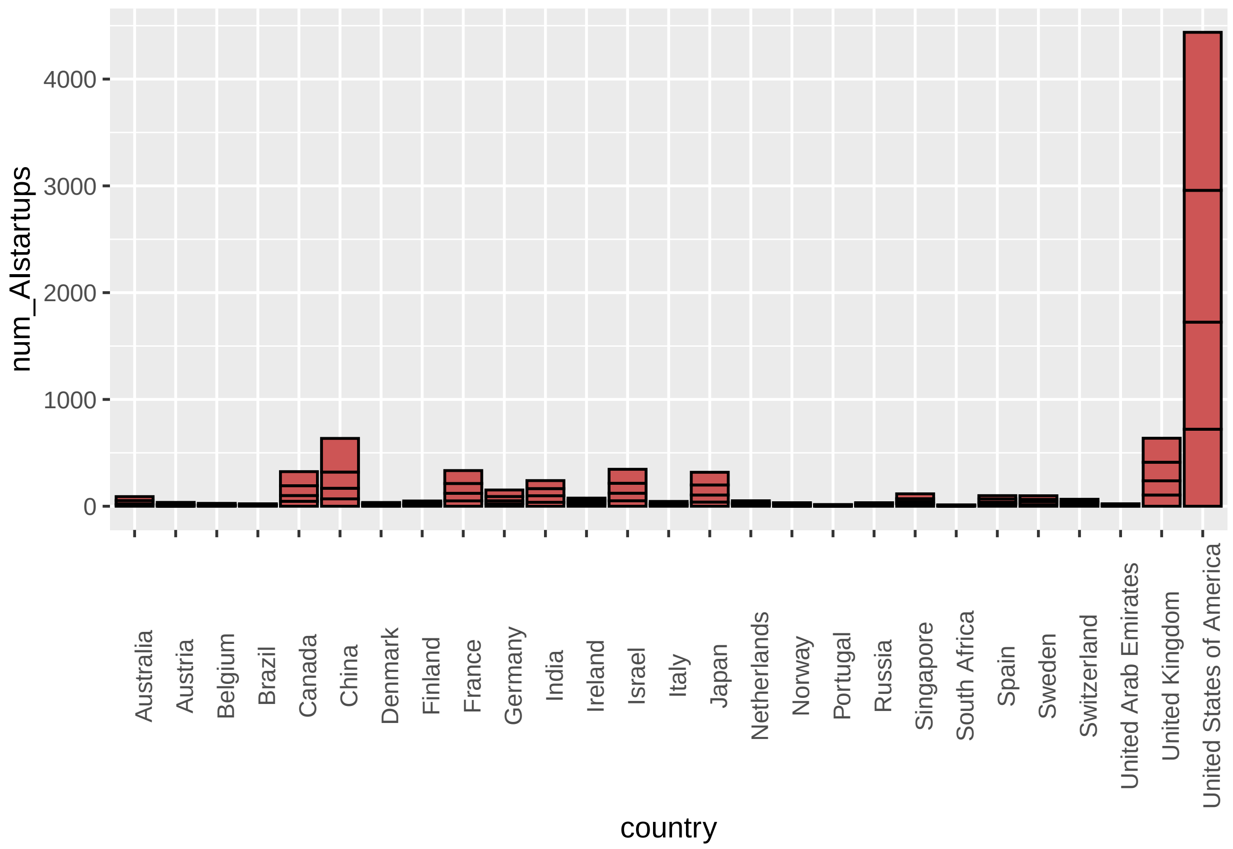
```
aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = country , y = female_ai_authors ) ) +
  geom_col( fill = "indianred3",
              color = "black" ) +
  theme(axis.text.x = element_text(angle = 90))
```



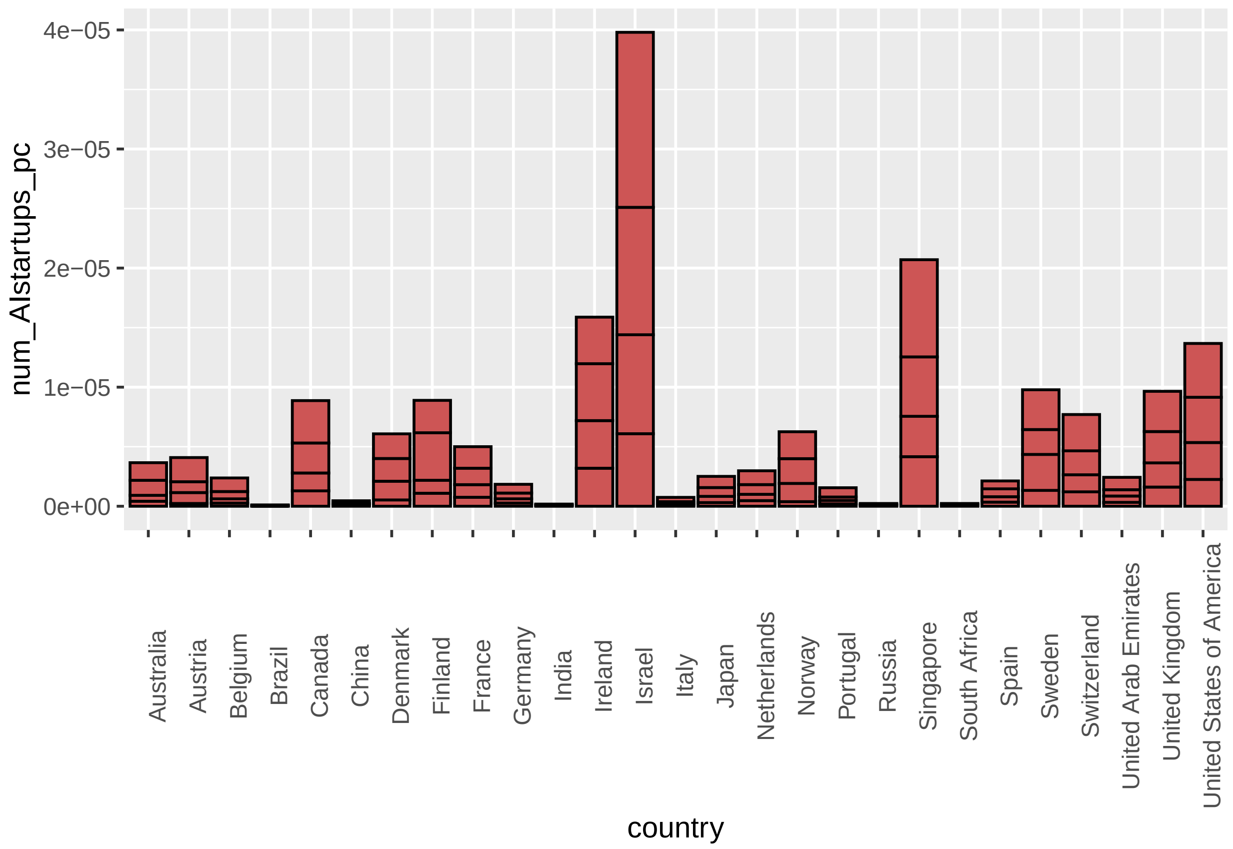
```
aiVibrancyIndicators2019.tib %>%
  filter(year == "2018") %>%
  ggplot( aes( x = country , y = female_ai_skill_penetration) ) +
  geom_col( fill = "indianred3",
              color = "black" ) +
  theme(axis.text.x = element_text(angle = 90))
```



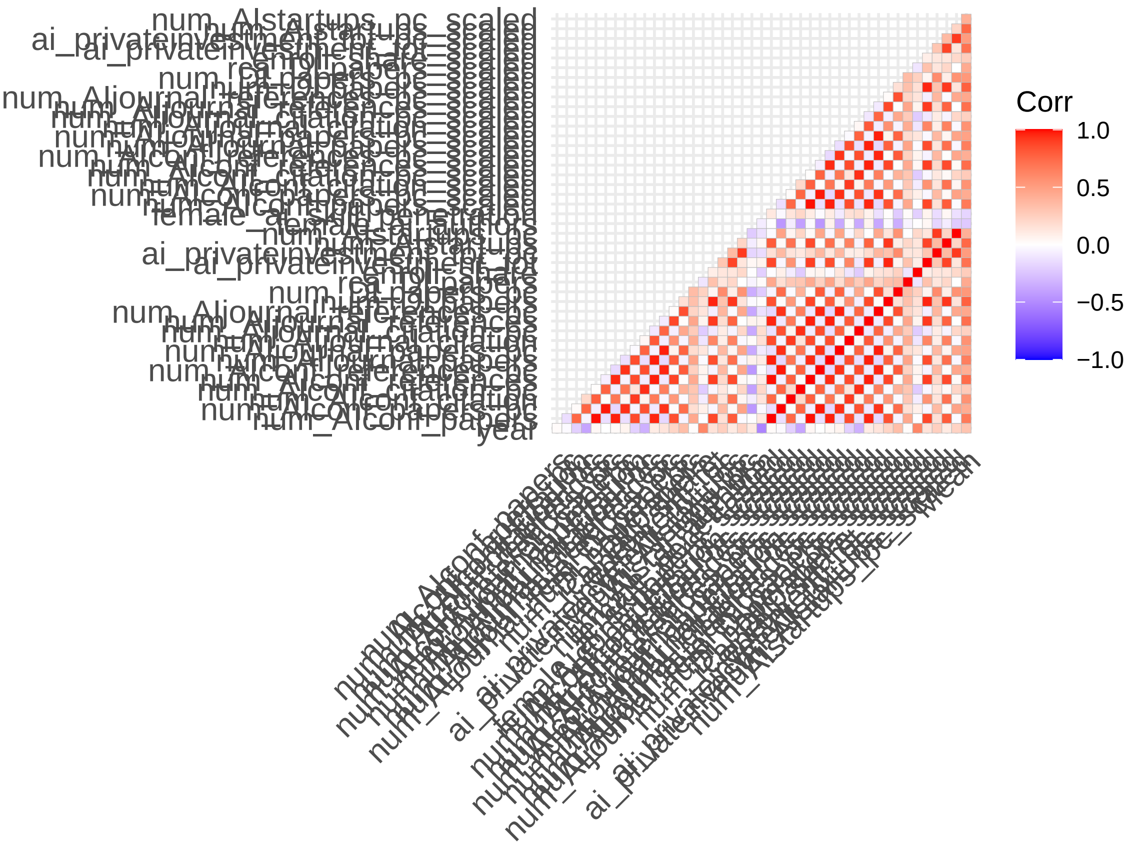
```
aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = country , y = num_AIstartups ) ) +
  geom_col( fill = "indianred3",
              color = "black" ) +
  theme(axis.text.x = element_text(angle = 90))
```



```
aiVibrancyIndicators2019.tib %>%
  ggplot( aes( x = country , y = num_AIstartups_pc ) ) +
  geom_col( fill = "indianred3",
              color = "black" ) +
  theme(axis.text.x = element_text(angle = 90))
```



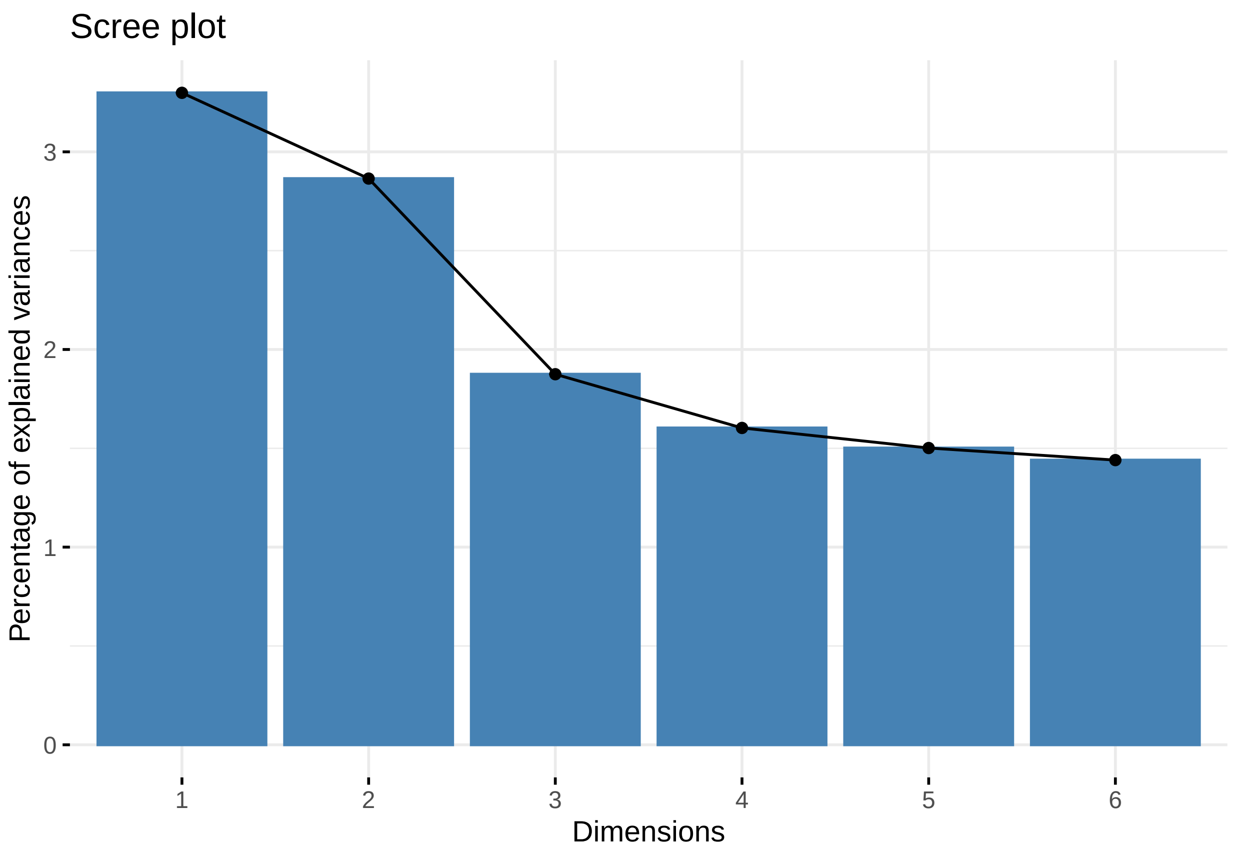
```r
aiVibrancyNumeric.tib <- aiVibrancyIndicators2019.tib %>%
  select_if( is.numeric )

r.mat = cor( aiVibrancyNumeric.tib, use = "complete.obs" )

ggcorrplot( r.mat,
            type = "lower",
            insig = "blank" )
```



```
gower_dist <- daisy( aiVibrancyIndicators2019.tib %>% select( -country ),
                      metric = "gower",
                      type = list( logratio = 3 )
                    )

gower.summ <- summary( gower_dist )
gower_mat <- as.matrix(gower_dist)

closest.tib <-
aiVibrancyIndicators2019.tib[
        which( gower_mat == min(gower_mat[ gower_mat != min(gower_mat)]),
                arr.ind = TRUE)[1, ], ]

farthest.tib <-
aiVibrancyIndicators2019.tib[
      which(gower_mat == max(gower_mat[gower_mat != max(gower_mat)]),
              arr.ind = TRUE)[1, ], ]

k.vec <- 1:15

get_pam_silwidth <- function( k, dist )
{
  pam.clust <- pam( dist, diss=TRUE, k=k )
  return( pam.clust$silinfo$avg.width )
}
```

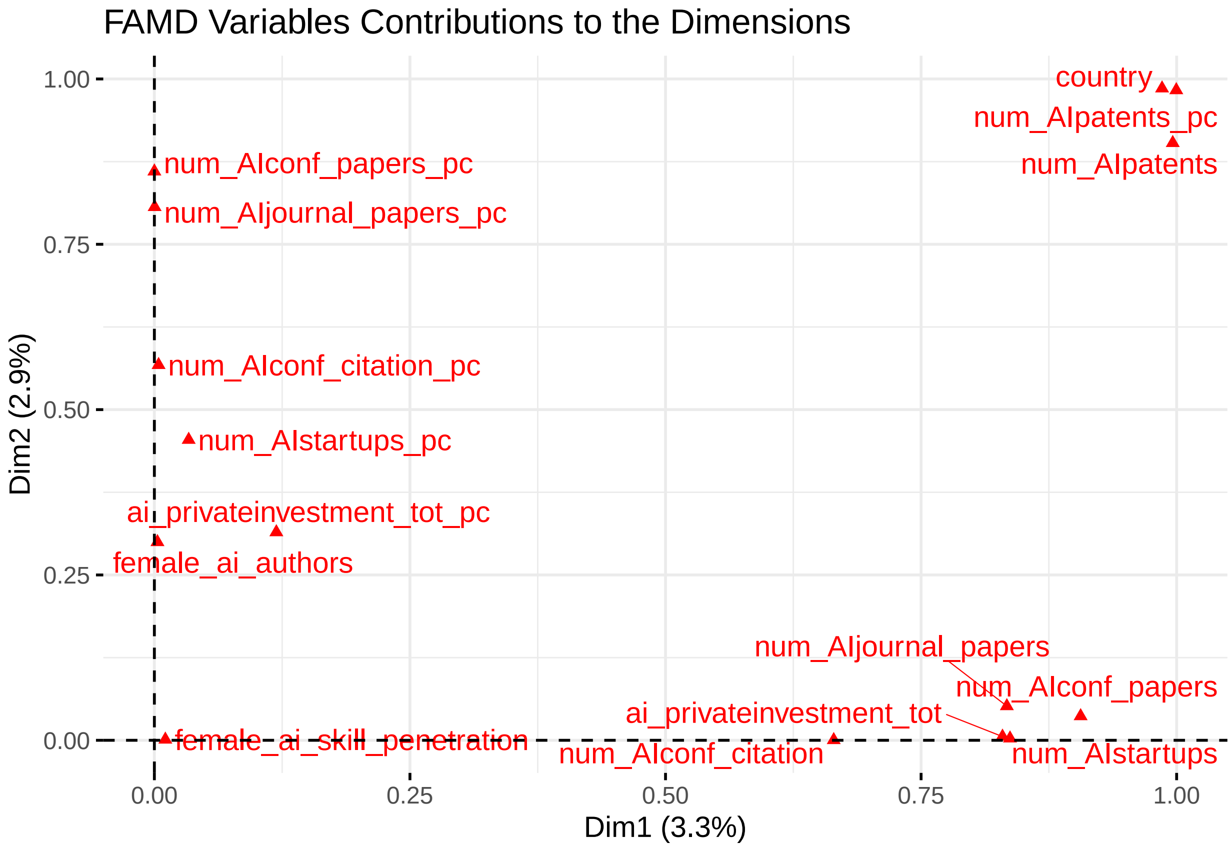
```
sil_width.vec <- map_dbl( k.vec[ -1 ],
  get_pam_silwidth,
dist=gower_dist
  )

pam_sil.tib <- tibble( k = k.vec,
  sil = c( 0, sil_width.vec )
)

sil_max <- with( pam_sil.tib,
  which( sil == max( sil ) )
)

pam.clust <- pam( x = gower_dist,
  k = sil_max,
  diss = TRUE
)

aiVibrancyIndicators2019.tib <- aiVibrancyIndicators2019.tib %>%
    mutate(p3 = factor( paste0( 'p', pam.clust$clustering ) ) )

aiVibrancyIndicators2019_vars.vec <-
  c("country", "num_AIconf_papers","num_AIconf_papers_pc", "num_AIconf_citation", "num_AIconf_citation_p

aiVibrancyIndicators2019.famd <-
  FAMD( aiVibrancyIndicators2019.tib %>%
          select(all_of(aiVibrancyIndicators2019_vars.vec)), ncp = 6, graph = FALSE )

( aiVibrancyIndicators2019.scree.ggplot <-
  fviz_screeplot( aiVibrancyIndicators2019.famd ) %>%
  labs( title = "Scree Plot") )
```

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

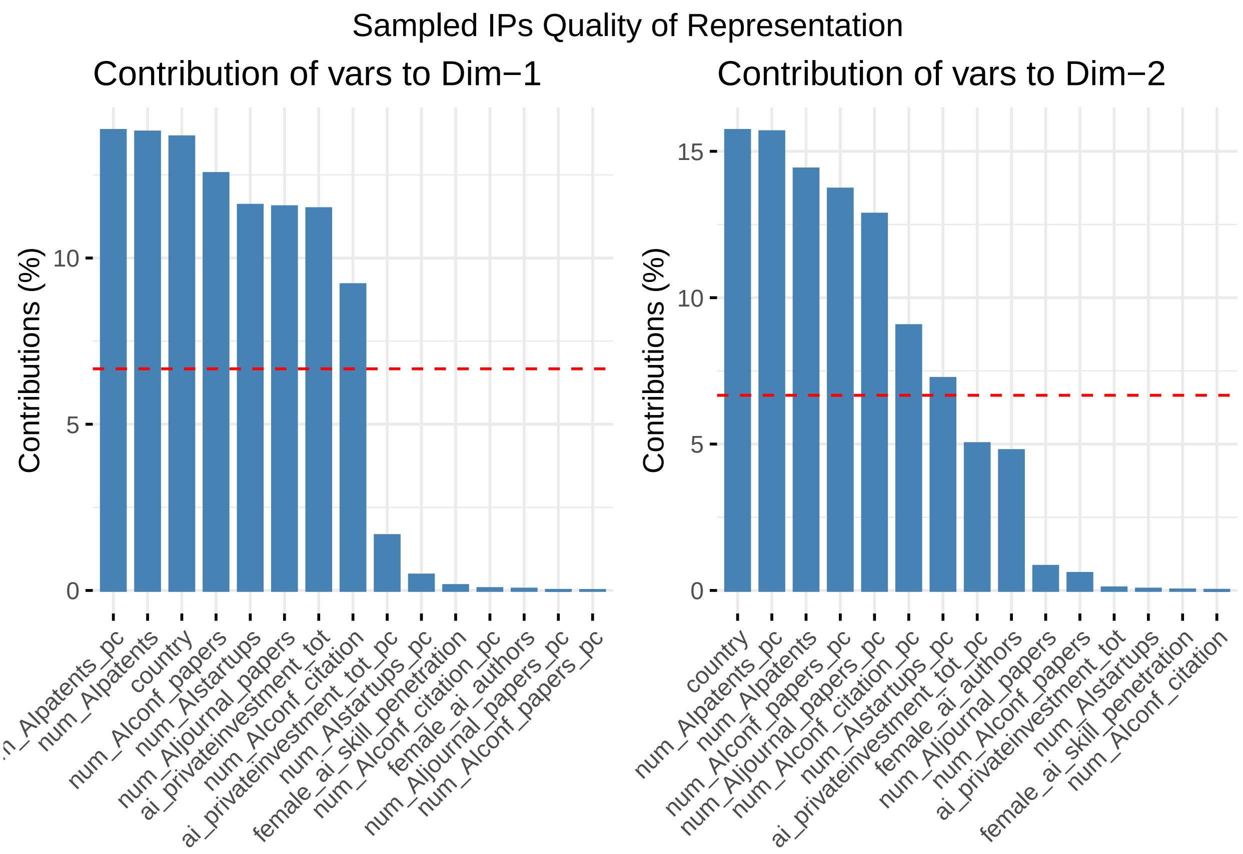


```
##
## $title
## [1] "Scree Plot"
##
## attr(,"class")
## [1] "labels"
```

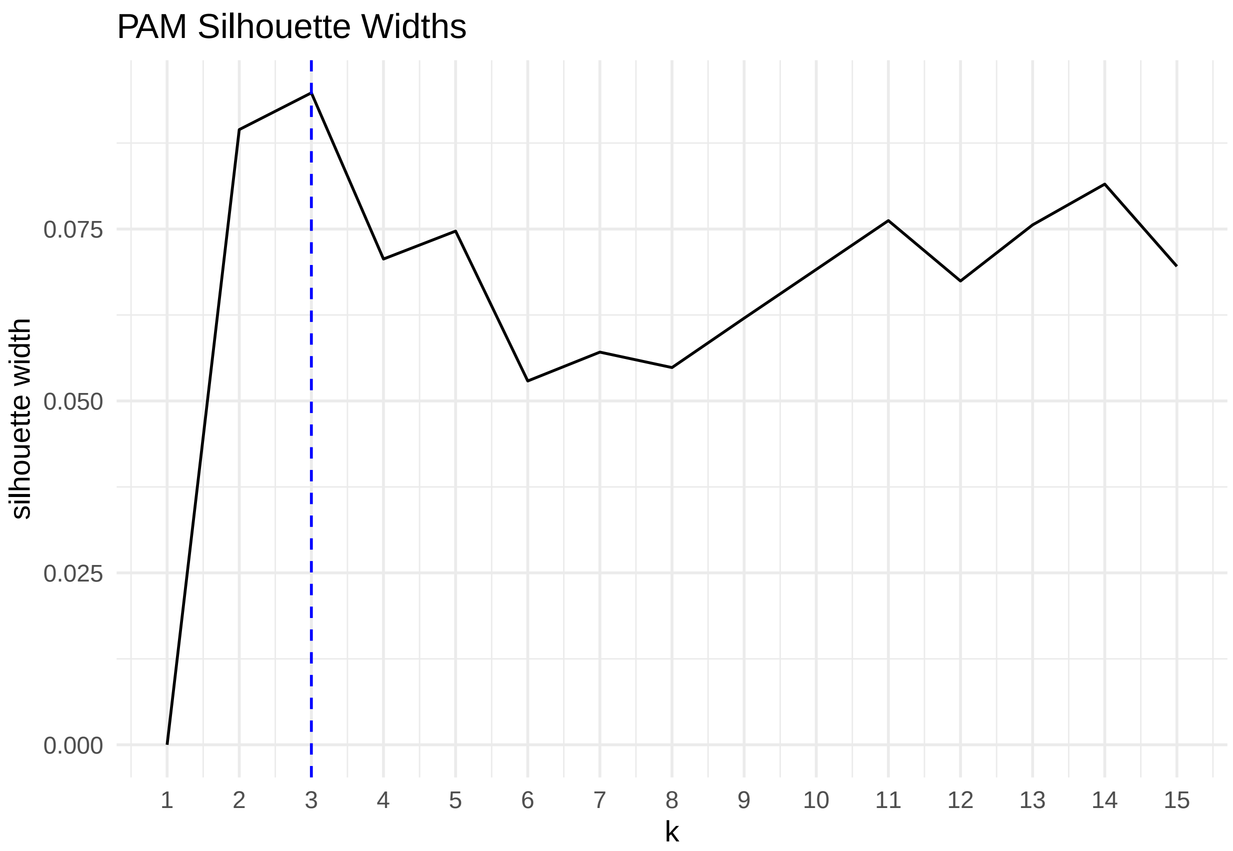
```
( aiVibrancyIndicators2019.var.ggplot <-
      fviz_famd_var( aiVibrancyIndicators2019.famd, repel = TRUE ) +
      labs(title = "FAMD Variables Contributions to the Dimensions"))
```



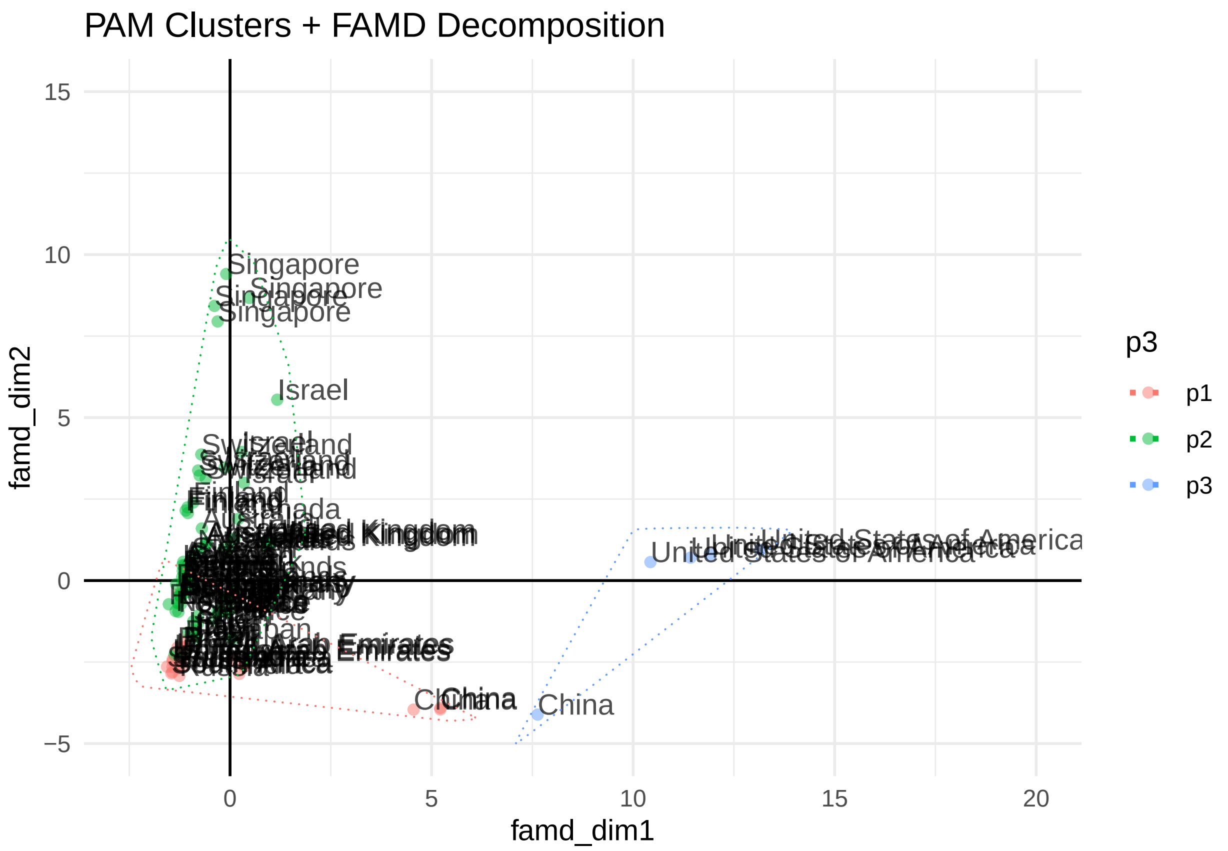
```
aiVibrancyIndicators2019.contrib.dim1.ggplot <-
      fviz_contrib( aiVibrancyIndicators2019.famd, "var", axes = 1,
                    title = "Contribution of vars to Dim-1" )

aiVibrancyIndicators2019.contrib.dim2.ggplot <-
      fviz_contrib( aiVibrancyIndicators2019.famd, "var", axes = 2,
                    title = "Contribution of vars to Dim-2" )

contrib.grid <-
    grid.arrange(aiVibrancyIndicators2019.contrib.dim1.ggplot,
                 aiVibrancyIndicators2019.contrib.dim2.ggplot, nrow = 1,
              top = "Sampled IPs Quality of Representation")
```



```r
( pam_sil.ggplot <- pam_sil.tib %>%
  ggplot( mapping = aes( x = k, y = sil ) ) +
              geom_line() +
              geom_vline( xintercept = sil_max,
              linetype="dashed", color="blue" ) +
              theme_minimal() +
              scale_x_continuous( breaks = k.vec ) +
              labs( title = "PAM Silhouette Widths" ) +
              ylab( "silhouette width" ) )
```



```r
aiVibrancyIndicators2019.tib <- aiVibrancyIndicators2019.tib %>%
  mutate( famd_dim1 = aiVibrancyIndicators2019.famd$ind$coord[, 1],
          famd_dim2 = aiVibrancyIndicators2019.famd$ind$coord[, 2]
        )

pam_clusters_guide = "PAM\nClusters"

pam_cluster_colors.vec <- brewer.pal( sil_max, name="Set1" )
names(pam_cluster_colors.vec) <- paste0( 'p', 1:sil_max )

quali_ind.tib <- tibble( var  = rownames( aiVibrancyIndicators2019.famd$quali.var$coord ),
                         dim1 = aiVibrancyIndicators2019.famd$quali.var$coord[, 1],
                         dim2 = aiVibrancyIndicators2019.famd$quali.var$coord[, 2]
                       )

(aiVibrancyIndicators2019.ggplot <- aiVibrancyIndicators2019.tib %>%
  ggplot( mapping = aes( x = famd_dim1, y = famd_dim2) ) +
    geom_vline( xintercept = 0 ) +
    geom_hline( yintercept = 0 ) +
    geom_point( mapping = aes( color = p3 ),
                alpha = .5 ) +
    geom_text(aes(label= country),hjust=0, vjust=0, alpha = .7) +
    geom_encircle( mapping = aes( group = p3, color = p3 ),
                   linetype = "dotted", s_shape = 0.95 ) +
    theme_minimal() +
    coord_cartesian( xlim = c(-2.5, 20), ylim = c(-5, 15) ) +
    labs( title = "PAM Clusters + FAMD Decomposition" ) )
```


PAM Clusters + FAMD Decomposition
famd_dim2
famd_dim1
p3
p1
p2
p3
Singapore
Singapore
Singapore
Singapore
Israel
Israel
Israel
Switzerland
Switzerland
Finland
Canada
Australia
United Kingdom
Netherlands
Germany
France
Japan
United Arab Emirates
South Africa
China
China
China
United States of America
United States of America
United States of America