Nonparametric Project of Agricultural Productivity in the U.S.

Sequence Clustering and Permutational Manova

Sofia Moroni*

2023-06-27

Contents

1 Load Libraries

```r
library(TraMineR)
```

```
## Warning: il pacchetto 'TraMineR' è stato creato con R versione 4.1.3
```

```
##
## TraMineR stable version 2.2-7 (Built: 2023-04-17)
```

```
## Website: http://traminer.unige.ch
```

```
## Please type 'citation("TraMineR")' for citation information.
```

```r
library(dtw)
```

```
## Warning: il pacchetto 'dtw' è stato creato con R versione 4.1.3
```

```
## Caricamento del pacchetto richiesto: proxy
```

```
##
## Caricamento pacchetto: 'proxy'
```

*sofia.moroni@mail.polimi.it

```
## I seguenti oggetti sono mascherati da 'package:stats':
##
##     as.dist, dist

## Il seguente oggetto è mascherato da 'package:base':
##
##     as.matrix

## Loaded dtw v1.23-1. See ?dtw for help, citation("dtw") for use in publication.
```

```
library(cluster)
library(ggplot2)
```

```
## Warning: il pacchetto 'ggplot2' è stato creato con R versione 4.1.3
```

```
library(maps)
```

```
## Warning: il pacchetto 'maps' è stato creato con R versione 4.1.3

##
## Caricamento pacchetto: 'maps'

## Il seguente oggetto è mascherato da 'package:cluster':
##
##     votes.repub
```

```
library(mapdata)
```

```
## Warning: il pacchetto 'mapdata' è stato creato con R versione 4.1.3
```

```
library(usmap)
library(dplyr)
```

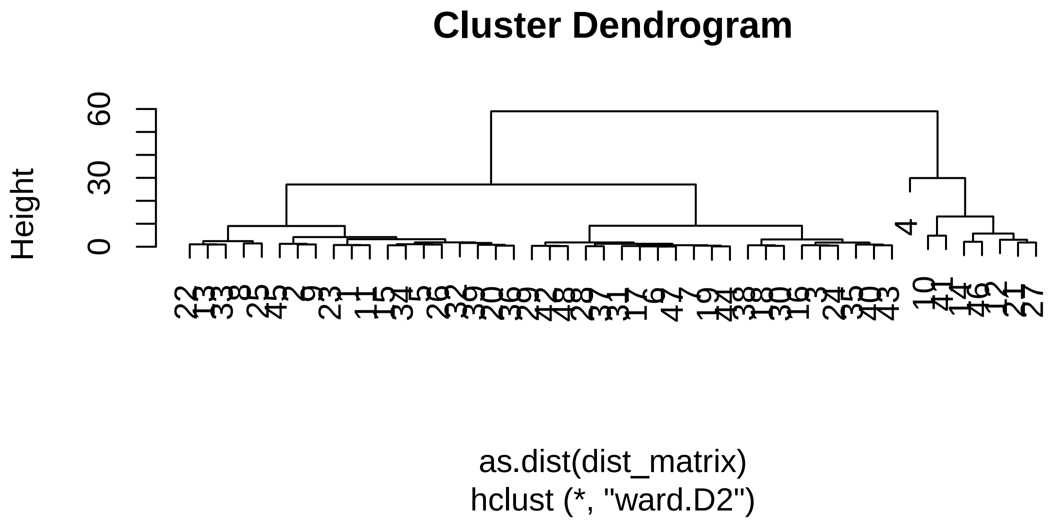
```
## Warning: il pacchetto 'dplyr' è stato creato con R versione 4.1.3

##
## Caricamento pacchetto: 'dplyr'

## I seguenti oggetti sono mascherati da 'package:stats':
##
##     filter, lag

## I seguenti oggetti sono mascherati da 'package:base':
##
##     intersect, setdiff, setequal, union
```

2 Load data

```
data_path = file.path('data')
output_path = file.path('results')
data =
    read.table(
        file.path(data_path, 'total_output_by_states.csv'),
        header = T,
        sep = ';'
    )

data = data[1:45,]

# Sostituzione delle virgole con punti
data<- data.frame(lapply(data, function(x) gsub(",", ".", x)))
data <- as.data.frame(lapply(data, as.numeric))

data = t(data)
colnames(data) <- data[1, ]
data <- data[-1, ]
data = as.data.frame(data)
state = rownames(data)
```

3 Sequence Clustering

```
data2 =data

dist_matrix <- matrix(NA, nrow = nrow(data2), ncol = nrow(data2))

for (i in 1:nrow(data2)) {
  for (j in i:nrow(data2)) {
    dtw_distance <- dtw(data2[i, ], data2[j, ])$distance
    dist_matrix[i, j] <- dtw_distance
    dist_matrix[j, i] <- dtw_distance
  }
}

cluster_results <- hclust(as.dist(dist_matrix), method = "ward.D2")
plot(cluster_results)
```



```
# Specify the number of clusters you want
num_clusters <- 4  # Adjust as needed

# Extract cluster assignments
cluster_labels <- cutree(cluster_results, k = num_clusters)
```

```
#cluster_labels = c(cluster_labels[1:3],4,cluster_labels[4:47])
state <- map_data("state")
states = unique(state$region)[-8]

cluster_data =data.frame(cluster_labels,state =states)
#d= merge(state,cluster_data, by.x = "region", by.y = "state")
```

PREPARE DATA

Capital Input

```
capital_input =
    read.table(
        file.path(data_path, 'capital_input.csv'),
        header = T,
        sep = ';'
    )

# Sostituzione delle virgole con punti
capital_input<- data.frame(lapply(capital_input, function(x) gsub(",", ".", x)))
capital_input <- as.data.frame(lapply(capital_input, as.numeric))
capital_input =t(capital_input)
colnames(capital_input) <- capital_input[1, ]
capital_input <- capital_input[-1, ]
capital_input = as.data.frame(capital_input)

capital_med <- apply(capital_input, MARGIN = 1, FUN = median)
cluster_data = data.frame(capital_input =capital_med,cluster_data)
```

Labor Input

```r
labor_input =
    read.table(
        file.path(data_path, 'labor_input_by_states.csv'),
        header = T,
        sep = ';'
    )

# Sostituzione delle virgole con punti
labor_input<- data.frame(lapply(labor_input, function(x) gsub(",", ".", x)))
labor_input <- as.data.frame(lapply(labor_input, as.numeric))
labor_input =t(labor_input)
colnames(labor_input) <- labor_input[1, ]
labor_input <- labor_input[-1, ]
labor_input = as.data.frame(labor_input)

labor_med <- apply(labor_input, MARGIN = 1, FUN = median)
cluster_data = data.frame(labor_input =labor_med,cluster_data )
```

Intermediate Input

```r
intermediate_input =
    read.table(
        file.path(data_path, 'total_intermediate_input_by_states.csv'),
        header = T,
        sep = ';'
    )

# Sostituzione delle virgole con punti
intermediate_input<- data.frame(lapply(intermediate_input, function(x) gsub(",", ".", x)))
intermediate_input <- as.data.frame(lapply(intermediate_input, as.numeric))
intermediate_input =t(intermediate_input)
colnames(intermediate_input) <- intermediate_input[1, ]
intermediate_input <- intermediate_input[-1, ]
intermediate_input = as.data.frame(intermediate_input)
intermediate_input = intermediate_input[1:48,]

intermediate_med <- apply(intermediate_input, MARGIN = 1, FUN = median)
cluster_data= data.frame(intermediate_input =intermediate_med,cluster_data )
```

Total Output

```r
total_output =
    read.table(
        file.path(data_path, 'total_output_by_states.csv'),
        header = T,
        sep = ';'
    )

total_output = total_output[1:45,]

# Sostituzione delle virgole con punti
```

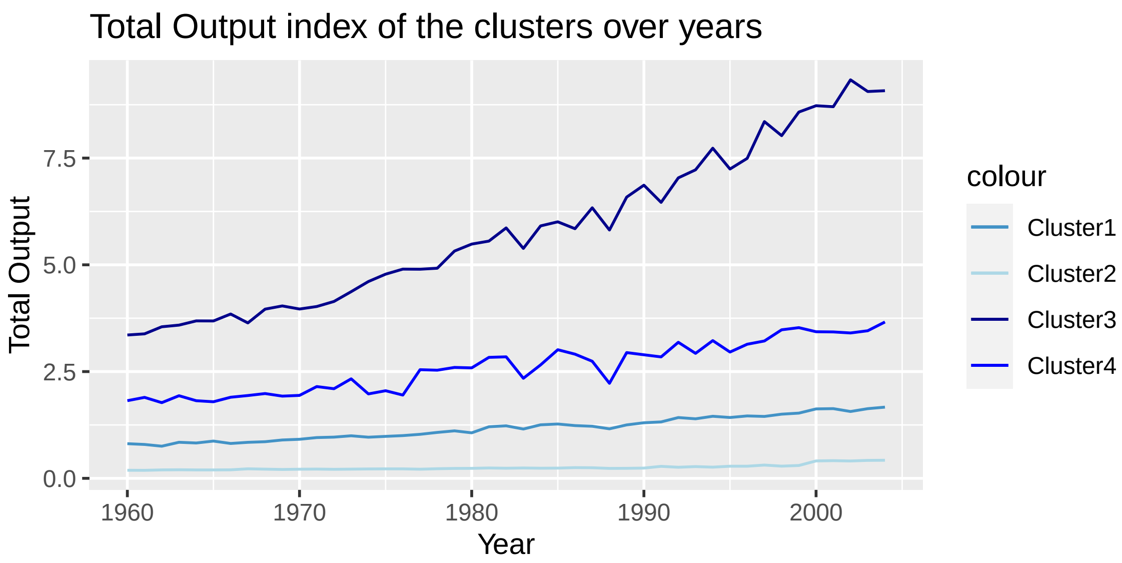
```
total_output<- data.frame(lapply(total_output, function(x) gsub(",", ".", x)))
total_output <- as.data.frame(lapply(total_output, as.numeric))
total_output =t(total_output)
colnames(total_output) <- total_output[1, ]
total_output <- total_output[-1, ]
total_output = as.data.frame(total_output)
total_output = total_output[1:48,]

output_med <- apply(total_output, MARGIN = 1, FUN = median)
cluster_data = data.frame(total_output =output_med,cluster_data )
```

Plot over years

```
total_output$cluster_label = cluster_labels

output_med <- total_output %>%
  group_by(cluster_label) %>%
  summarize(across(starts_with("19"), median, na.rm = TRUE),across(starts_with("20"), mean, na.rm = TRU
```

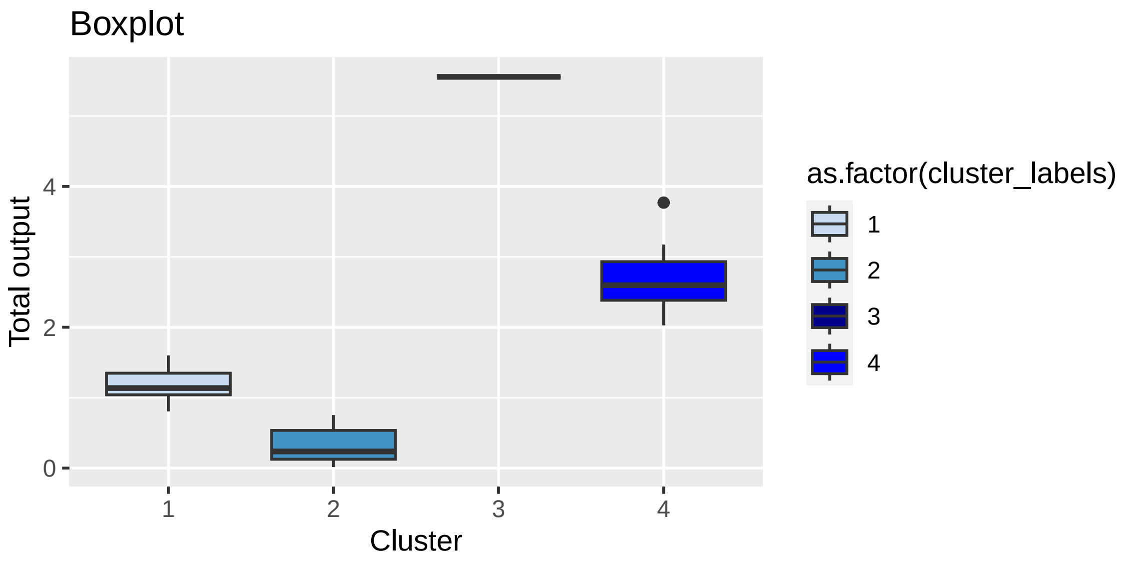
```
## Warning: There was 1 warning in ‘summarize()‘.
## i In argument: ‘across(starts_with("19"), median, na.rm = TRUE)‘.
## i In group 1: ‘cluster_label = 1‘.
## Caused by warning:
## ! The ‘...‘ argument of ‘across()‘ is deprecated as of dplyr 1.1.0.
## Supply arguments directly to ‘.fns‘ through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```
data_plot =t(output_med)
colnames(data_plot) = data_plot[1,]
data_plot = data_plot[-1,]
data_plot = as.data.frame(data_plot)
data_plot$Year =as.numeric(seq(1960,2004))
colnames(data_plot) = c("Cluster1","Cluster2","Cluster3","Cluster4","Year")
rownames(data_plot) = seq(1:45)


ggplot(data_plot, aes(x = Year)) +
  geom_line(aes(y = Cluster1, color = "Cluster1")) +
  geom_line(aes(y = Cluster2, color = "Cluster2")) +
  geom_line(aes(y = Cluster3 ,color = "Cluster3")) +
  geom_line(aes(y = Cluster4, color = "Cluster4")) +
  labs(title = "Total Output index of the clusters over years ",
       x = "Year", y = "Total Output") +
  scale_color_manual(values = c("Cluster1" =  "#4292C6", "Cluster2" = "lightblue",
                                "Cluster3" = "darkblue", "Cluster4" = "blue"  ))
```



```
index1 =which(cluster_data$cluster_labels==1)
index2 =which(cluster_data$cluster_labels==2)
index3 =which(cluster_data$cluster_labels==3)
index4 =which(cluster_data$cluster_labels==4)
```

```
# BOXPLOT rispetto ai clusters
ggplot(cluster_data, aes(x = as.factor(cluster_labels), y = total_output, fill = as.factor(cluster_label
  geom_boxplot() +
  scale_fill_manual(values = c("1" = "#C6DBEF", "2" = "#4292C6", "3" = "darkblue", "4"="blue")) +
  labs(title = "Boxplot", x = "Cluster", y = "Total output")
```



4 MANOVA

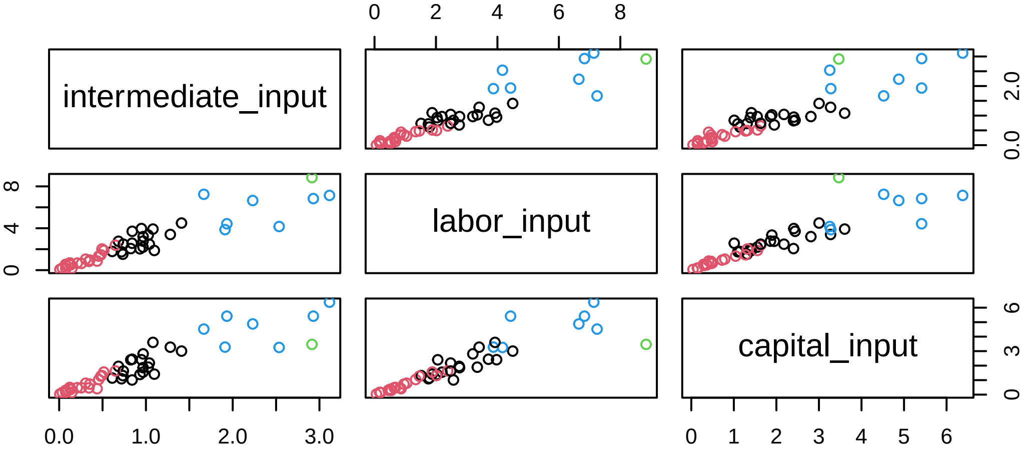
```
fit <- manova(as.matrix(cluster_data[,2:4]) ~ cluster_labels)
summary.manova(fit,test="Wilks")
```

```
##                 Df   Wilks approx F num Df den Df    Pr(>F)
## cluster_labels  1 0.66147   7.5061      3     44 0.0003668 ***
## Residuals      46
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
T0 <- -summary.manova(fit,test="Wilks")$stats[1,2]
T0
```

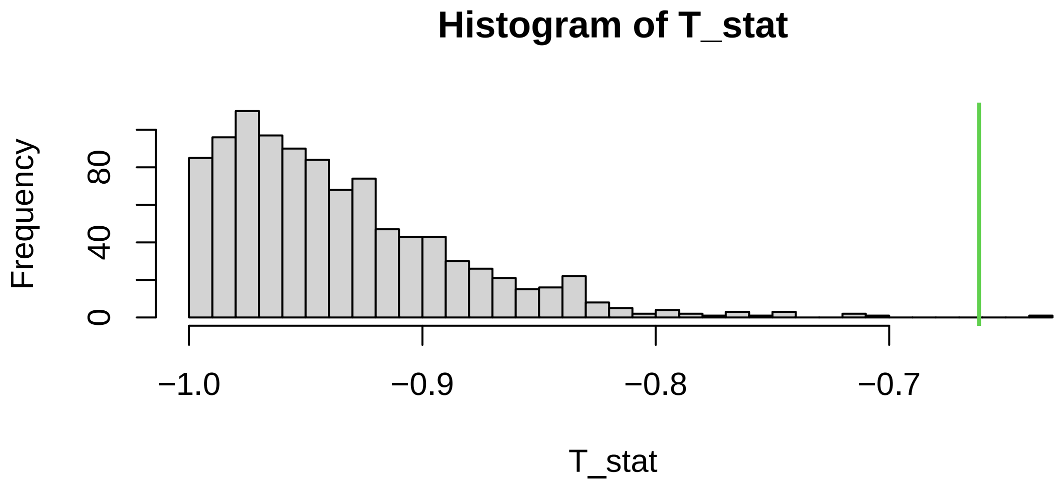
```
## [1] -0.6614721
```

```
plot(cluster_data[,2:4],col=cluster_labels)
```

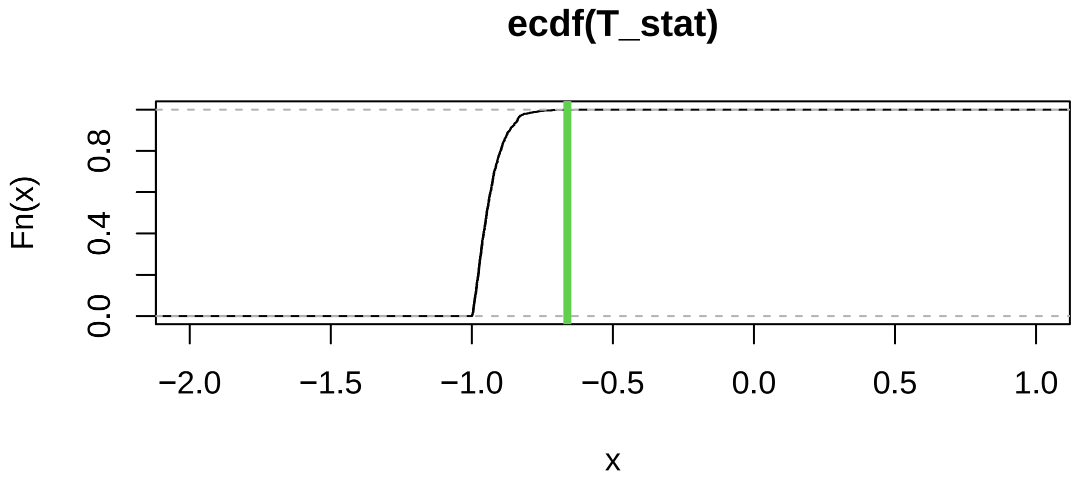


```
set.seed(100)
B=1000
T_stat <- numeric(B)
n =48

for(perm in 1:B){
  # choose random permutation
  permutation <- sample(1:n)
  cluster_labels.perm <- cluster_labels[permutation]
  fit.perm <- manova(as.matrix(cluster_data[,2:4]) ~ cluster_labels.perm)
  T_stat[perm] <- -summary.manova(fit.perm,test="Wilks")$stats[1,2]
}
```

```
hist(T_stat,xlim=range(c(T_stat,T0)),breaks=30)
abline(v=T0,col=3,lwd=2)
```



```
plot(ecdf(T_stat),xlim=c(-2,1))
abline(v=T0,col=3,lwd=4)
```



```
# p-value
p_val <- sum(T_stat>=T0)/B
p_val
```

```
## [1] 0.001
```

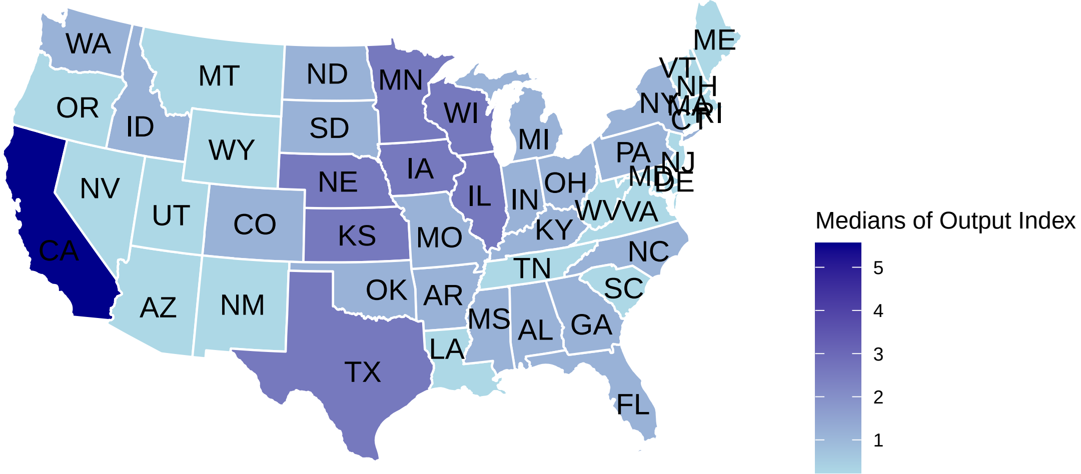
#PLOTS

```r
capital_input$cluster_label = cluster_labels
capital_med <- capital_input %>%
  group_by(cluster_label) %>%
  summarize(across(starts_with("19"), median, na.rm = TRUE),across(starts_with("20"), mean, na.rm = TRU

labor_input$cluster_label = cluster_labels
labor_med <- labor_input %>%
  group_by(cluster_label) %>%
  summarize(across(starts_with("19"), median, na.rm = TRUE),across(starts_with("20"), mean, na.rm = TRU

intermediate_input$cluster_label = cluster_labels
intermediate_med <- intermediate_input %>%
  group_by(cluster_label) %>%
  summarize(across(starts_with("19"), median, na.rm = TRUE),across(starts_with("20"), mean, na.rm = TRU
```

```r
#Calcolo per ogni cluster la mediana sugli anni dell'output, e degli input
output = apply(output_med[,-1],1,median)
capital = apply(capital_med[,-1],1,median)
labor = apply(labor_med[,-1],1,median)
intermediate = apply(intermediate_med[,-1],1,median)

medians = matrix(nrow =48,ncol=4)
medians[index1,] =matrix(c(output[1],capital[1],labor[1],intermediate[1]), nrow = length(index1), ncol =
medians[index2,] =matrix(c(output[2],capital[2],labor[2],intermediate[2]), nrow = length(index2), ncol =
medians[index3,] =matrix(c(output[3],capital[3],labor[3],intermediate[3]), nrow = length(index3), ncol =
medians[index4,] =matrix(c(output[4],capital[4],labor[4],intermediate[4]), nrow = length(index4), ncol =

State = rownames(data)

medians=as.data.frame(medians)
medians$state = State
colnames(medians)=c("output","capital","labor","intermediate","state")
```

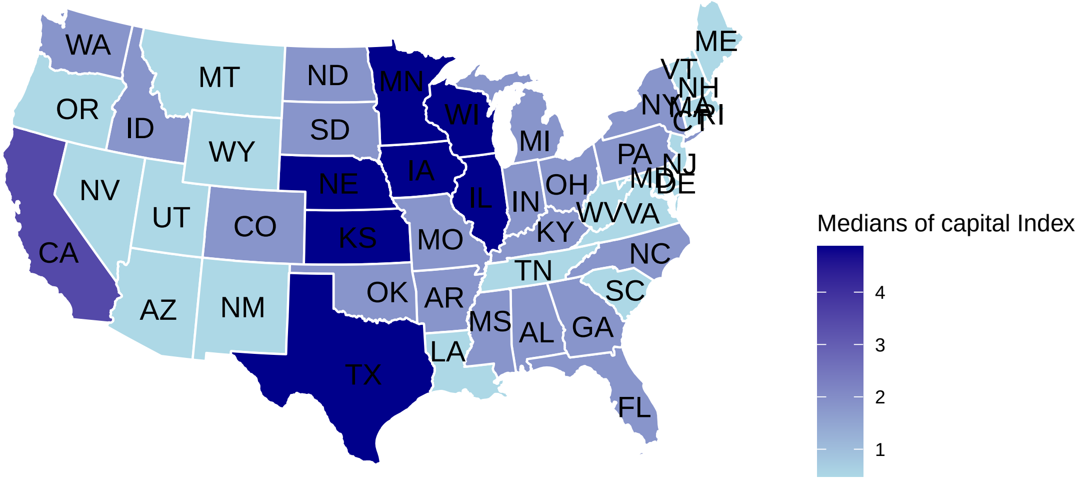
OUTPUT

```r
library(usmap)
library(ggplot2)
State = rownames(cluster_data)

plot_usmap(regions = "state",include = State , data = medians, values = "output", color = "white",label
    scale_fill_continuous(low = "lightblue", high = "darkblue",name = "Medians of Output Index", label =
    theme(legend.position = "right")
```



CAPITAL

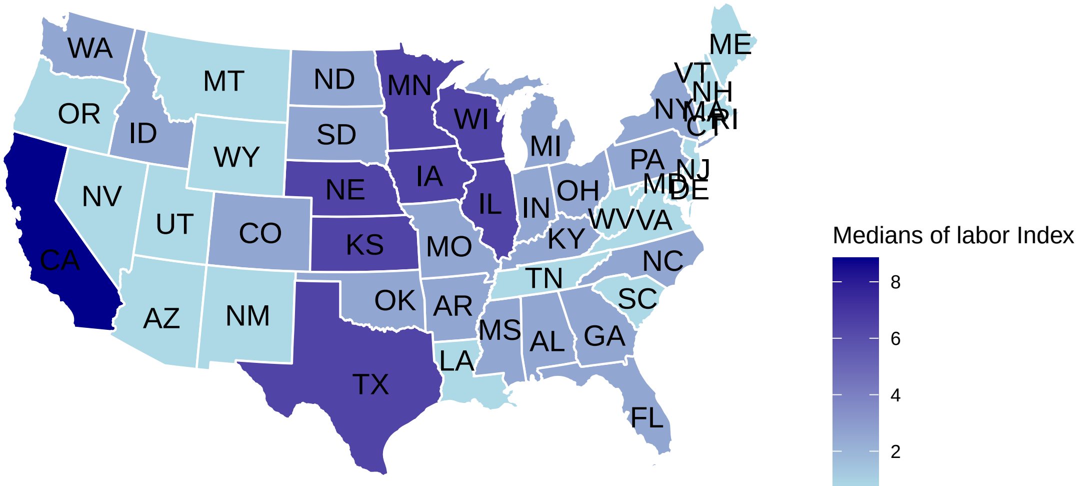
```
library(usmap)
library(ggplot2)
State = rownames(cluster_data)

plot_usmap(regions = "state",include = State , data = medians, values = "capital", color = "white",labe
    scale_fill_continuous(low = "lightblue", high = "darkblue",name = "Medians of capital Index", label
    theme(legend.position = "right")
```



LABOR

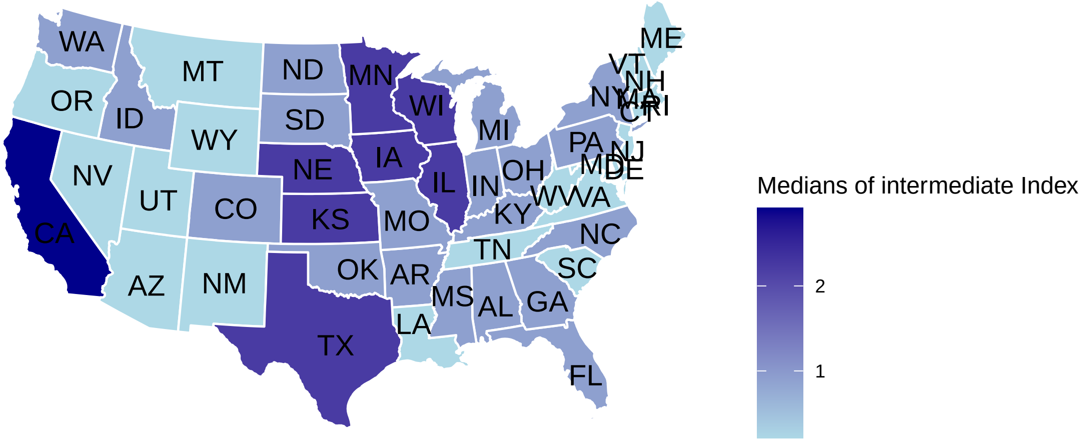
```
library(usmap)
library(ggplot2)
```

```
State = rownames(cluster_data)

plot_usmap(regions = "state",include = State , data = medians, values = "labor", color = "white",labels
    scale_fill_continuous(low = "lightblue", high = "darkblue",name = "Medians of labor Index", label =
    theme(legend.position = "right")
```



INTERMEDIATE

```
library(usmap)
library(ggplot2)
State = rownames(cluster_data)

plot_usmap(regions = "state",include = State , data = medians, values = "intermediate", color = "white"
    scale_fill_continuous(low = "lightblue", high = "darkblue",name = "Medians of intermediate Index",
    theme(legend.position = "right")
```


WA
MT
ND
MN
ME
VT
NH
OR
ID
SD
WI
NY
MA
CT
RI
MI
WY
PA
NJ
NV
NE
IA
MD
DE
IL
IN
OH
UT
CO
KY
WV
VA
CA
KS
MO
NC
TN
AZ
NM
OK
AR
SC
MS
AL
GA
LA
TX
FL
Medians of intermediate Index
2
1