# Moral Machines in the EU

David Steenmeijer

2022/05/02

## Getting started

This notebook guides the reader through the R code, used in the Master's thesis 'Moral Machines in the EU: How Civilian Preferences Can Help Shape Policy'. Created in partial fulfillment for the Master's Data Science & Society at Tilburg University.

First, the relevant packages must be imported:

```
# Import packages
library('dplyr') # for data manipulation
library('ggplot2') # for data visualization
library('tidyr') # for tidying data
library('stats') # for running statistical tests and clustering
library('dendextend') # for custom visualization of dendrograms
library('colorspace') # custom visualization
library('mclust') # plotting two dendrograms in one
library('rstatix') #for get_summary_stats function
library('ggstatsplot') # for anova's

## Set working directory and seed
setwd("~/Desktop/Scriptie/Thesis")
set.seed(1)
```

Next step is to load the data. My computer's memory was too small to load the data from the Moral Machine at once, which is why I used a CSV splitter to divide the data set up into 4 different sets. After selecting the right countries, I merged the data into one file:

```
## load Moral Machine Experiment data set as MM
## data source: Awad, E. (2021, October 10). Moral Machine. Retrieved from osf.io/3hvt2
#MM <- read.csv('data/SharedResponses.csv', header = TRUE, sep = ',')

########### OR ###########
## load 4 splitted data sets and select EU member states
## MM_EU1
MM1 <- read.csv('data/SharedResponses1.csv', header = TRUE, sep = ',')
MM_EU1 <- MM1 %>%
                filter(UserCountry3 == 'AUT' | UserCountry3 == 'BEL' |
                        UserCountry3 == 'BGR' | UserCountry3 == 'HRV' |
                        UserCountry3 == 'CYP' | UserCountry3 == 'CZE' |
                        UserCountry3 == 'DNK' | UserCountry3 == 'EST' |
                        UserCountry3 == 'FIN' | UserCountry3 == 'FRA' |
                        UserCountry3 == 'DEU' | UserCountry3 == 'GRC' |
```

```r
                        UserCountry3 == 'HUN' | UserCountry3 == 'IRL' |
                        UserCountry3 == 'ITA' | UserCountry3 == 'LVA' |
                        UserCountry3 == 'LTU' | UserCountry3 == 'LUX' |
                        UserCountry3 == 'MLT' | UserCountry3 == 'NLD' |
                        UserCountry3 == 'POL' | UserCountry3 == 'PRT' |
                        UserCountry3 == 'ROU' | UserCountry3 == 'SVK' |
                        UserCountry3 == 'SVN' | UserCountry3 == 'ESP' |
                        UserCountry3 == 'SWE') %>%
                group_by(UserID) %>%
                group_by(UserCountry3)
write.csv(MM_EU1, "~Desktop/Scriptie/Thesis/MM_EU//MM_EU1.csv", row.names = TRUE)

##MM_EU2
MM2 <- read.csv('data/SharedResponses2.csv', header = TRUE, sep = ',')
MM_EU2 <- MM2 %>%
                filter(UserCountry3 == 'AUT' | UserCountry3 == 'BEL' |
                        UserCountry3 == 'BGR' | UserCountry3 == 'HRV' |
                        UserCountry3 == 'CYP' | UserCountry3 == 'CZE' |
                        UserCountry3 == 'DNK' | UserCountry3 == 'EST' |
                        UserCountry3 == 'FIN' | UserCountry3 == 'FRA' |
                        UserCountry3 == 'DEU' | UserCountry3 == 'GRC' |
                        UserCountry3 == 'HUN' | UserCountry3 == 'IRL' |
                        UserCountry3 == 'ITA' | UserCountry3 == 'LVA' |
                        UserCountry3 == 'LTU' | UserCountry3 == 'LUX' |
                        UserCountry3 == 'MLT' | UserCountry3 == 'NLD' |
                        UserCountry3 == 'POL' | UserCountry3 == 'PRT' |
                        UserCountry3 == 'ROU' | UserCountry3 == 'SVK' |
                        UserCountry3 == 'SVN' | UserCountry3 == 'ESP' |
                        UserCountry3 == 'SWE') %>%
                group_by(UserID) %>%
                group_by(UserCountry3)
write.csv(MM_EU2, "~Desktop/Scriptie/Thesis/MM_EU//MM_EU2.csv", row.names = TRUE)

## MM_EU3
MM3 <- read.csv('data/SharedResponses3.csv', header = TRUE, sep = ',')
MM_EU3 <- MM3 %>%
                filter(UserCountry3 == 'AUT' | UserCountry3 == 'BEL' |
                        UserCountry3 == 'BGR' | UserCountry3 == 'HRV' |
                        UserCountry3 == 'CYP' | UserCountry3 == 'CZE' |
                        UserCountry3 == 'DNK' | UserCountry3 == 'EST' |
                        UserCountry3 == 'FIN' | UserCountry3 == 'FRA' |
                        UserCountry3 == 'DEU' | UserCountry3 == 'GRC' |
                        UserCountry3 == 'HUN' | UserCountry3 == 'IRL' |
                        UserCountry3 == 'ITA' | UserCountry3 == 'LVA' |
                        UserCountry3 == 'LTU' | UserCountry3 == 'LUX' |
                        UserCountry3 == 'MLT' | UserCountry3 == 'NLD' |
                        UserCountry3 == 'POL' | UserCountry3 == 'PRT' |
                        UserCountry3 == 'ROU' | UserCountry3 == 'SVK' |
                        UserCountry3 == 'SVN' | UserCountry3 == 'ESP' |
                        UserCountry3 == 'SWE') %>%
                group_by(UserID) %>%
                group_by(UserCountry3)
write.csv(MM_EU3, "~Desktop/Scriptie/Thesis/MM_EU//MM_EU3.csv", row.names = TRUE)
```

```r
##MM_EU4
MM4 <- read.csv('data/SharedResponses4.csv', header = TRUE, sep = ',')
MM_EU4 <- MM4 %>%
                filter(UserCountry3 == 'AUT' | UserCountry3 == 'BEL' |
                        UserCountry3 == 'BGR' | UserCountry3 == 'HRV' |
                        UserCountry3 == 'CYP' | UserCountry3 == 'CZE' |
                        UserCountry3 == 'DNK' | UserCountry3 == 'EST' |
                        UserCountry3 == 'FIN' | UserCountry3 == 'FRA' |
                        UserCountry3 == 'DEU' | UserCountry3 == 'GRC' |
                        UserCountry3 == 'HUN' | UserCountry3 == 'IRL' |
                        UserCountry3 == 'ITA' | UserCountry3 == 'LVA' |
                        UserCountry3 == 'LTU' | UserCountry3 == 'LUX' |
                        UserCountry3 == 'MLT' | UserCountry3 == 'NLD' |
                        UserCountry3 == 'POL' | UserCountry3 == 'PRT' |
                        UserCountry3 == 'ROU' | UserCountry3 == 'SVK' |
                        UserCountry3 == 'SVN' | UserCountry3 == 'ESP' |
                        UserCountry3 == 'SWE') %>%
                group_by(UserID) %>%
                group_by(UserCountry3)
write.csv(MM_EU4, "~Desktop/Scriptie/Thesis/MM_EU//MM_EU4.csv", row.names = TRUE)
```

```r
MM_EU1 <- read.csv('MM_EU/MM_EU1.csv', header = TRUE, sep = ',')
MM_EU2 <- read.csv('MM_EU/MM_EU2.csv', header = TRUE, sep = ',')
MM_EU3 <- read.csv('MM_EU/MM_EU3.csv', header = TRUE, sep = ',')
MM_EU4 <- read.csv('MM_EU/MM_EU4.csv', header = TRUE, sep = ',')

# Merge data sets
MM_EU_12 <- full_join(MM_EU1, MM_EU2)
MM_EU_34 <- full_join(MM_EU3, MM_EU4)
MM_EU_full <- full_join(MM_EU_12, MM_EU_34)

# Save fully merged data frame of all EU rows
#write.csv(MM_EU_full, "~/Desktop/Scriptie/Thesis/MM_EU//MM_EU_full.csv", row.names = TRUE)
```

## Data manipulation

Every dilemma is represented in two rows (the two options to choose from). They are paired via the variable 'ResponseID'. Incomplete dilemma's exist of only a single, unique 'ResponseID'. These cannot be interpreted without their counterparts and are therefore deleted from the data set.

```r
## Returns a data set that only holds the duplicates (necessary to make pairs)
MM_EU_pairs <- MM_EU_full %>%
  filter(duplicated(ResponseID) | duplicated(ResponseID, fromLast = TRUE)) %>%
  select(-X) %>%
  arrange(ResponseID)
#write.csv(MM_EU_pairs, "~/Desktop/Scriptie/Thesis/MM_EU//MM_EU_pairs/csv", row.names = TRUE)
```

Since every dilemma is represented in two rows, the following code is used to merge those situations. This results in the net choice people made in the Moral Machine experiment. It subtracts the sitation where people died from the situation where people surivived

```
## Split into two dataframes: one with survivors, one with dead
MM_EU_alive <- MM_EU_pairs %>%
  filter(Saved == 1) # these people were saved in the experiment
MM_EU_dead <- MM_EU_pairs %>%
  filter(Saved == 0)
## join the dataframes to have one row per situation
MM_EU_long <- inner_join(MM_EU_alive, MM_EU_dead, keep = TRUE, by = c('ResponseID'))

## Subtract dead from living to get net outcomes of situations
Final_Frame <- MM_EU_long %>%
  mutate(Man = Man.x - Man.y, ## Man.x comes from MM_EU_alive; Man.y from MM_EU_dead
         Woman = Woman.x - Woman.y, ## vice versa for all variables
         Pregnant = Pregnant.x - Pregnant.y,
         Stroller = Stroller.x - Stroller.y,
         OldMan = OldMan.x - OldMan.y,
         OldWoman = OldWoman.x - OldWoman.y,
         Boy = Boy.x - Boy.y,
         Girl = Girl.x - Girl.y,
         Homeless = Homeless.x - Homeless.y,
         LargeWoman = LargeWoman.x - LargeWoman.y,
         LargeMan = LargeMan.x - LargeMan.y,
         Criminal = Criminal.x - Criminal.y,
         MaleExecutive = MaleExecutive.x - MaleExecutive.y,
         FemaleExecutive = FemaleExecutive.x - FemaleExecutive.y,
         FemaleAthlete = FemaleAthlete.x - FemaleAthlete.y,
         MaleAthlete = MaleAthlete.x - MaleAthlete.y,
         FemaleDoctor = FemaleDoctor.x - FemaleDoctor.y,
         MaleDoctor = MaleDoctor.x - MaleDoctor.y,
         Dog = Dog.x - Dog.y,
         Cat = Cat.x - Cat.y) %>%
  rename(ResponseID = ResponseID.x, ## these variables are the same for both data sets, only one needed
         UserCountry3 = UserCountry3.x,
         Intervention = Intervention.x,
         DefaultChoiceIsOmission = DefaultChoiceIsOmission.x,
         DiffNumberOFCharacters = DiffNumberOFCharacters.x) %>%
  summarize(ResponseID, UserCountry3, ## selects the right variables for the data set with the net surv
            Intervention, DefaultChoiceIsOmission,
            DiffNumberOFCharacters,
            Man, Woman, Pregnant, Stroller, OldMan, OldWoman, Boy, Girl,
            Homeless, LargeWoman, LargeMan, Criminal, MaleExecutive, FemaleExecutive,
            FemaleAthlete, MaleAthlete, FemaleDoctor, MaleDoctor, Dog, Cat)

## removing 5 rows with only NA values (except for country and ID)
show(Final_Frame[1506155, ])
Final_Frame <- Final_Frame[-c(1786, 1506155, 1861222, 2411506, 3095031), ]

## Save new data frame as csv
write.csv(Final_Frame, '~/Desktop/Scriptie/Thesis/MM_EU//Final_Frame.csv')
```

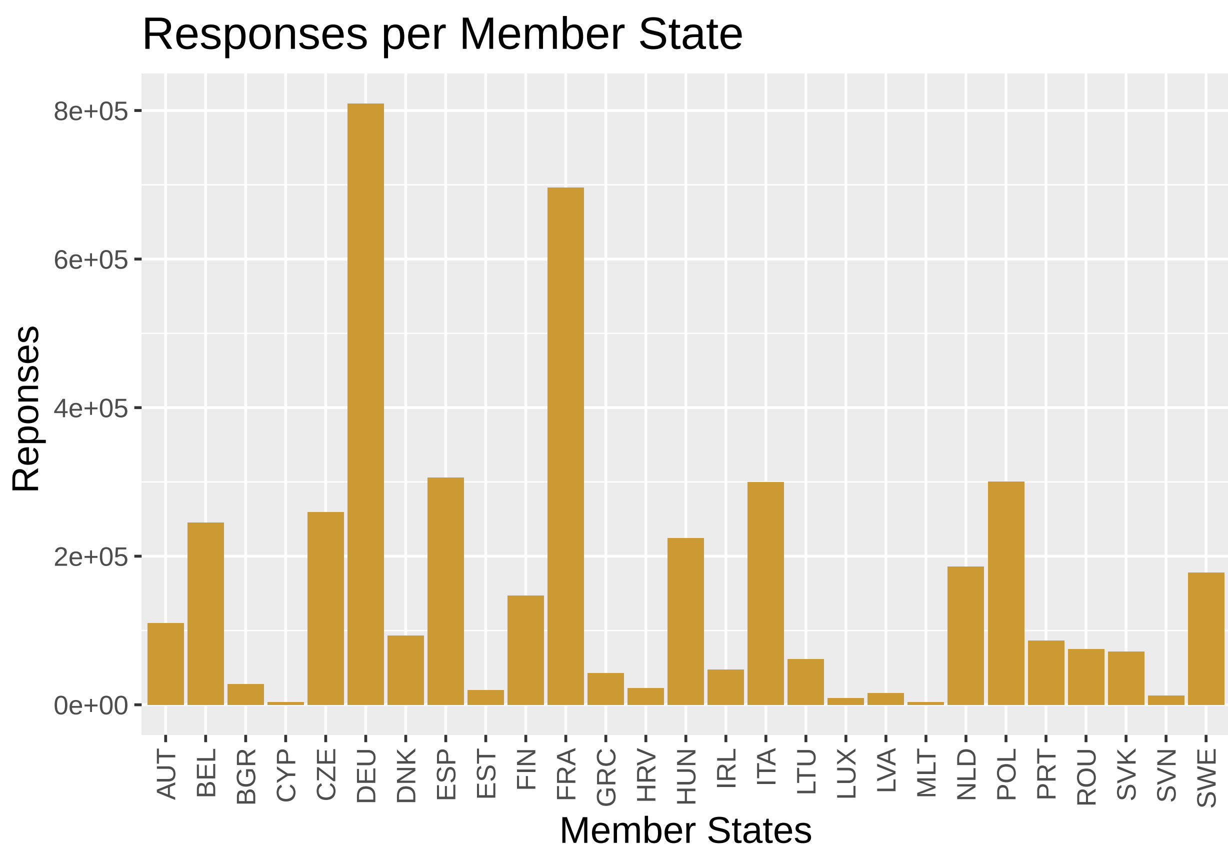
### Exploratory Data Analysis:

```
Final_Frame <- read.csv('MM_EU/Final_Frame.csv', header = TRUE, sep = ',')
```

```
## EU MEMBER STATES (27 STATES AS OF 2021)
##    AUSTRIA         = AUT
##    BELGIUM         = BEL
##    BULGARIA        = BGR
##    CROATIA         = HRV
##    CYPRUS          = CYP
##    CZECH REPUBLIC  = CZE
##    DENMARK         = DNK
##    ESTONIA         = EST
##    FINLAND         = FIN
##    FRANCE          = FRA
##    GERMANY         = DEU   MAXIMUM: 2.424.077
##    GREECE          = GRC
##    HUNGARY         = HUN
##    IRELAND         = IRL
##    ITALY           = ITA
##    LATVIA          = LVA
##    LITHUANIA       = LTU
##    LUXEMBURG       = LUX
##    MALTA           = MLT   MINIMUM: 10.597
##    NETHERLANDS     = NLD
##    POLAND          = POL
##    PORTUGAL        = PRT
##    ROMANIA         = ROU
##    SLOVAKIA        = SVK
##    SLOVENIA        = SVN
##    SPAIN           = ESP
##    SWEDEN          = SWE

## EDA sanity check
head(Final_Frame)
str(Final_Frame)
dim(Final_Frame)
```

**Responses per Member State** This plot shows the distribution of responses per Member State Germany and France are very well represented, whereas smaller countries like Cyprus and Luxembourg only have a couple thousand responses

```
## Plots number of responses per Member State
ggplot(Final_Frame, aes(x = UserCountry3)) +
  geom_bar(stat = 'count', position = 'dodge', fill = '#CC9933') +
  scale_x_discrete(name = "Member States") +
  scale_y_continuous(name = "Reponses ") +
  theme(legend.position = "bottom", text = element_text(size = 14),
        axis.text = element_text(size = 10)) +
  labs(title = 'Responses per Member State') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



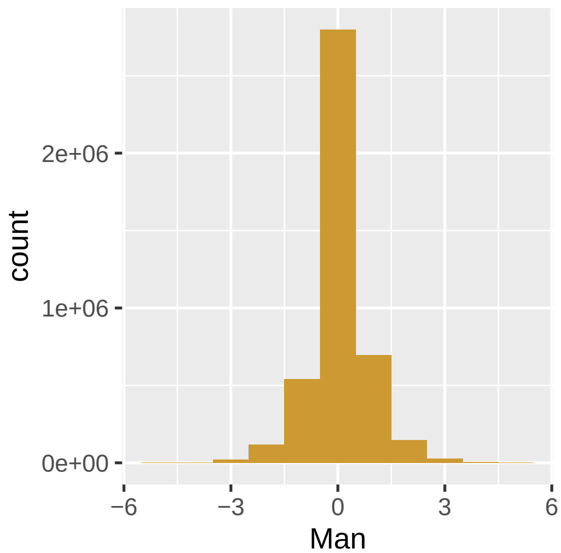

```
count(Final_Frame, UserCountry3) %>% arrange((n)) # shows number of rows per country
```

```
##    UserCountry3      n
## 1           MLT   3532
## 2           CYP   4007
## 3           LUX   8906
## 4           SVN  12317
## 5           LVA  15849
## 6           EST  19566
## 7           HRV  22776
## 8           BGR  27728
## 9           GRC  42592
## 10          IRL  47611
## 11          LTU  61600
## 12          SVK  72020
## 13          ROU  74942
## 14          PRT  86212
## 15          DNK  93071
## 16          AUT 110155
## 17          FIN 147390
## 18          SWE 177860
## 19          NLD 185920
## 20          HUN 224745
## 21          BEL 245369
## 22          CZE 259221
## 23          ITA 299759
## 24          POL 300324
```

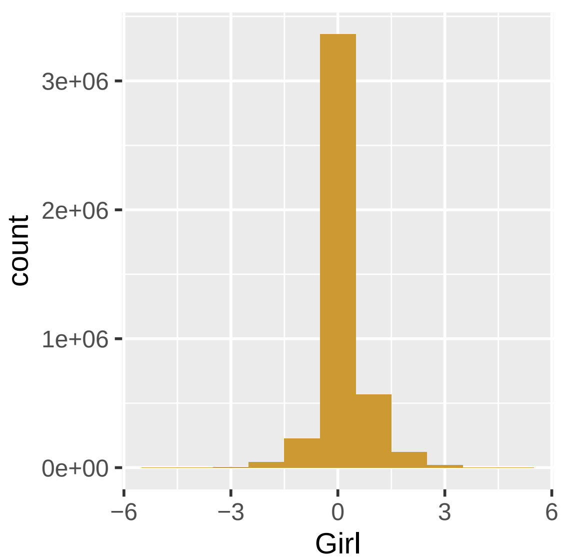
```
## 25          ESP 306044
## 26          FRA 696297
## 27          DEU 809467
```

**Net survivors** Some more EDA with the new data frame of net survivors

```
## look at some of the distributions
ggplot(Final_Frame, aes(Man)) +
  geom_histogram(bins = 11, fill = '#CC9933')
```



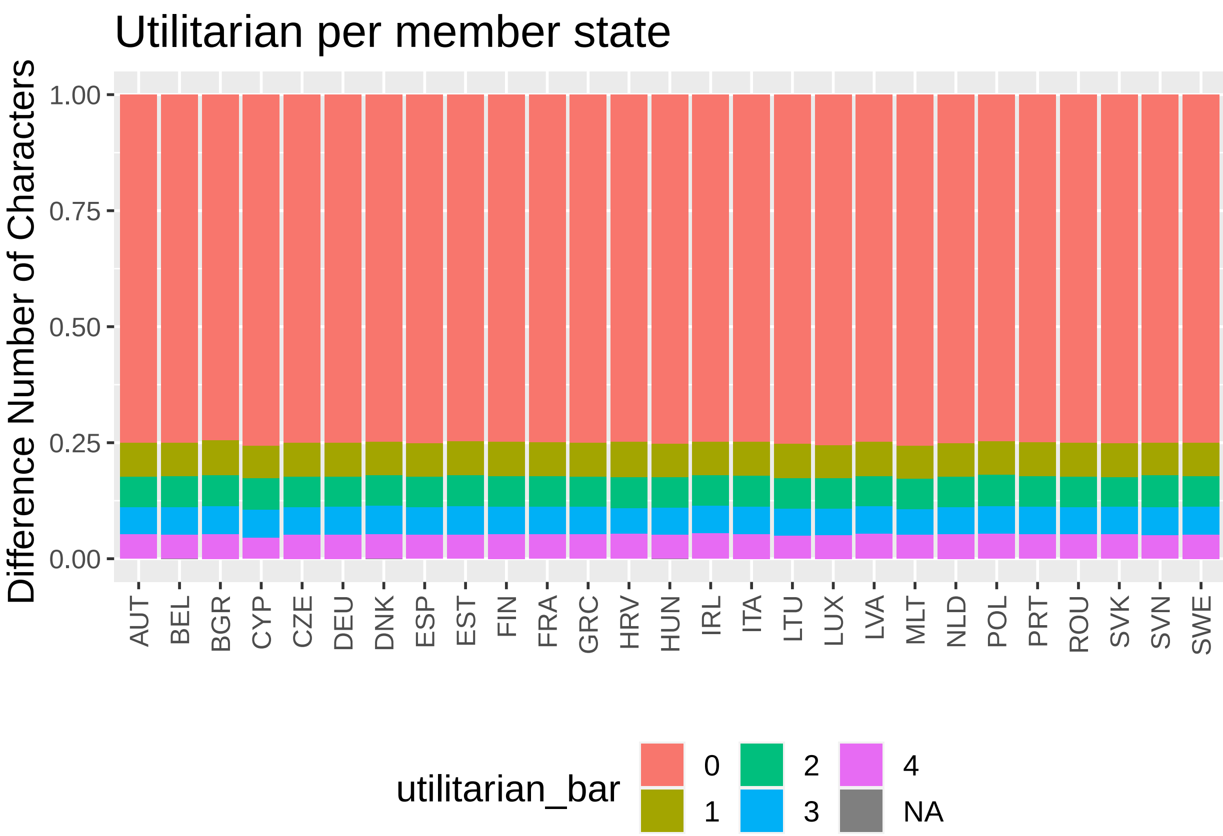

```
ggplot(Final_Frame, aes(Girl)) +
         geom_histogram(bins = 11, fill = '#CC9933')
```




**Utilitarianism**

The following plot shows the distribution of the 'Utilitarian' variable per Member State A visual difference would indicate more or less utilitarian preferences The numbers represent the number of people saved per situation

```r
## Plots Difference in Number of Characters per country
utilitarian_bar <- as.character(Final_Frame$DiffNumberOFCharacters,
                                levels = c(4, 3, 2, 1, 0))
utilitarian_plt <- ggplot(Final_Frame, aes(x = UserCountry3, fill = utilitarian_bar)) +
  geom_bar(stat = 'count', position = 'fill') +
  scale_x_discrete(name = "") +
  scale_y_continuous(name = "Difference Number of Characters ") +
  theme(legend.position = "bottom", text = element_text(size = 14),
        axis.text = element_text(size = 10)) +
  labs(title = 'Utilitarian per member state') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
utilitarian_plt
```




As the plot indicates, there is no visual difference noticeable. This indicates that the countries have a homogeneous preference for saving more people.

### Gender

In order to get more insight in the differences per gender, the data is manipulated, to combine al gendered variables to plot the differences per gender, controlled for age, fitness, and social status. The T-test shows that the means are significantly different from each other.

```r
### combines all gendered attributes to look only at difference gender
gender_total <- Final_Frame %>%
```

```r
  mutate(MenTotal = Man + OldMan + Boy + LargeMan + MaleExecutive + MaleAthlete + MaleDoctor,
         WomenTotal = Woman + OldWoman + Girl + LargeWoman + FemaleExecutive + FemaleAthlete + FemaleDoc
  summarise(ResponseID, UserCountry3, Intervention, DefaultChoiceIsOmission, DiffNumberOFCharacters, Men
## Gender: Welch Two Sample t-test: mean_men = 0.175, mean_women = 0.338, p-value = 2.2e-16
t.test(gender_total$MenTotal, gender_total$WomenTotal)
```

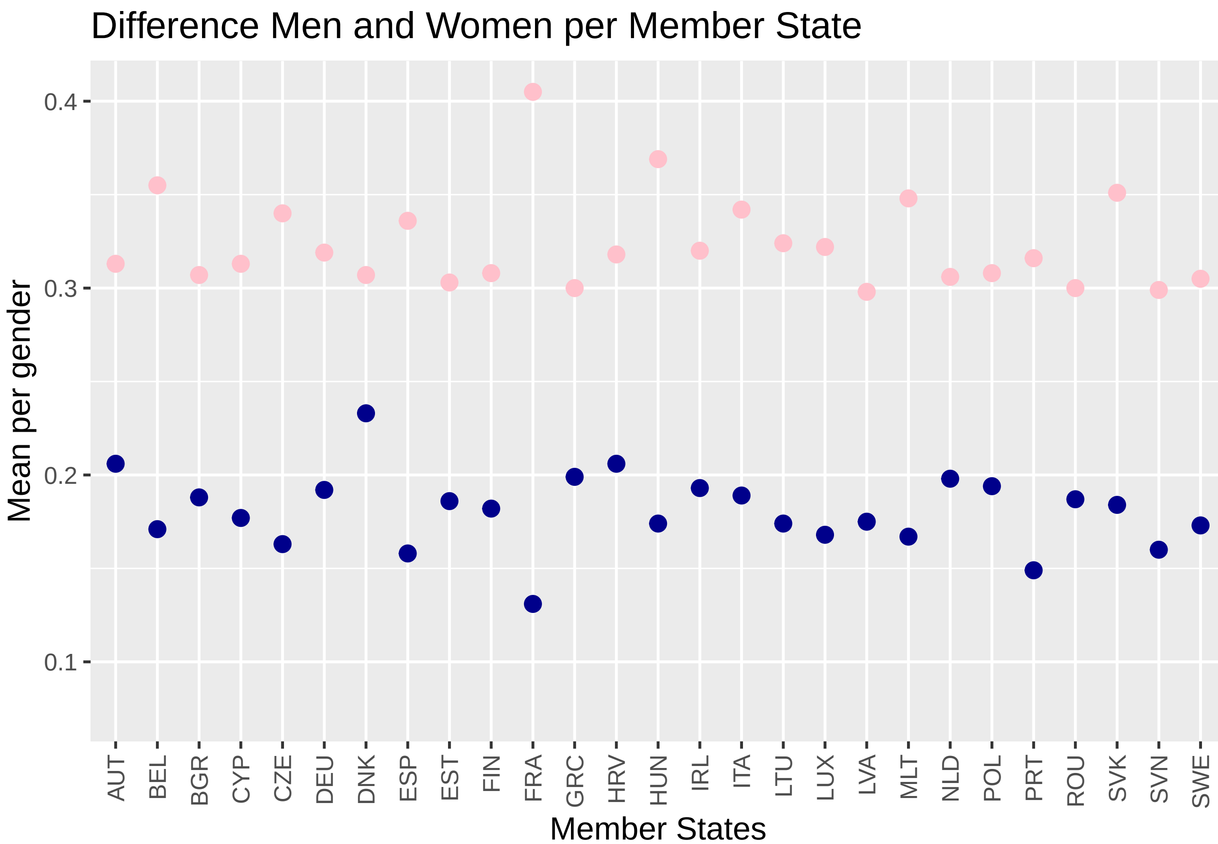
```
##
##  Welch Two Sample t-test
##
## data:  gender_total$MenTotal and gender_total$WomenTotal
## t = -145.97, df = 8708483, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1658551 -0.1614602
## sample estimates:
## mean of x mean of y
## 0.1748406 0.3384983
```

```r
## creates sets to look at difference in means women/men per country
women_per_country <- gender_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(WomenTotal, type = 'mean_sd') %>%
  arrange(UserCountry3)
men_per_country <- gender_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(MenTotal, type = 'mean_sd') %>%
  arrange(UserCountry3)
gender_per_country <- full_join(men_per_country, women_per_country, by = c('UserCountry3' = 'UserCountry
  rename(mean_men = mean.x,
         sd_men = sd.x,
         mean_women = mean.y,
         sd_women = sd.y) %>%
  mutate(difference = mean_women -  mean_men) %>%
  arrange(desc(difference)) %>%
  select(UserCountry3, n, mean_men, mean_women, difference, sd_men, sd_women) %>%
  arrange(difference)

order <- gender_per_country %>% arrange(difference) %>% summarise(UserCountry3)
order = as.list(order)
```

**Difference in gender per Member State** The following plot shows the difference in means between men and women, showing a real difference in gender preferences in all countries, with a maximum (and outlier) in France, and a minimum in Denmark. The difference here is the distance between the two dots on the dotplot.

```r
## plots gender differences per country
## Both men and women!!
ggplot(gender_per_country, aes(x = UserCountry3)) +
  geom_point(aes(y = mean_men), color = 'dark blue', size = 2.5) +
  geom_point(aes(y = mean_women), color = 'pink', size = 2.5) +
  geom_line(aes(y = difference)) +
  ylim(0.1, 0.45) +
```

```
  scale_x_discrete(name = "Member States") +
  scale_y_continuous(name = "Mean per gender") +
  theme(legend.position = "right", text = element_text(size = 12),
        axis.text = element_text(size = 9)) +
  labs(title = 'Difference Men and Women per Member State') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```




**Age**

The same data manipulation is done for age: Again, the T-tests show that the means are significantly different from each other

```
### combines all ages
ages_total <- Final_Frame %>%
  mutate(Young = (Boy + Girl + Stroller + Pregnant)/4, # weighed variables 4:2:2
         Man_Woman = (Man + Woman)/2,
         Old = (OldMan + OldWoman)/2) %>%
  summarise(ResponseID, UserCountry3, DiffNumberOFCharacters, Young, Old, Man_Woman)
## Age: Welch Two Sample t-test: mean_young = 0.310, mean_old = -0.154, p-value < 2.2e-16
t.test(ages_total$Young, ages_total$Old)
```

```
##
##  Welch Two Sample t-test
##
## data:  ages_total$Young and ages_total$Old
```

```
## t = 511.88, df = 5819185, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1536357 0.1548167
## sample estimates:
##   mean of x   mean of y
##  0.07745286 -0.07677334
```

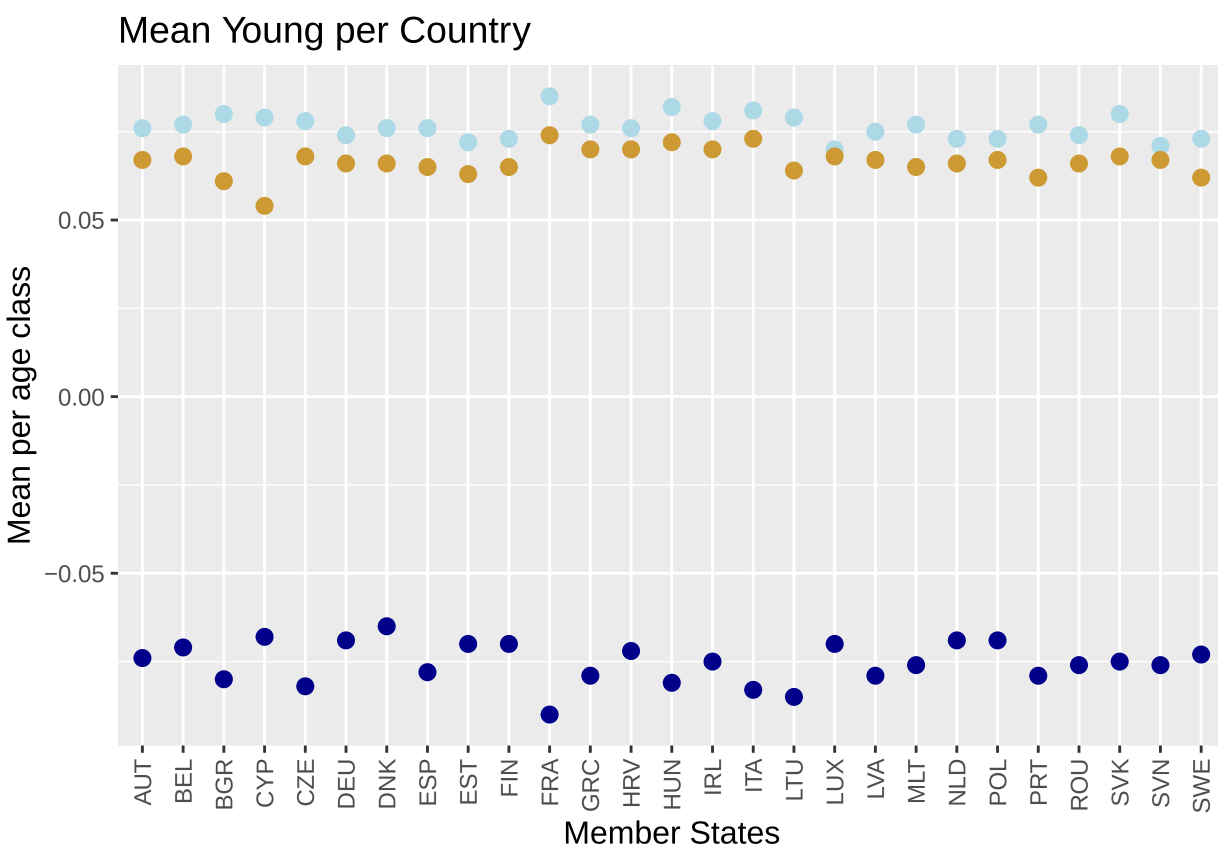
```
## Age versus norm: Welch Two Sample t-test: mean_Man_Woman = 0.136 mean_young = 0.310, p-value < 2.2e-
t.test(ages_total$Man_Woman, ages_total$Young)
```

```
##
##  Welch Two Sample t-test
##
## data:  ages_total$Man_Woman and ages_total$Young
## t = -32.468, df = 6011372, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.009810742 -0.008693715
## sample estimates:
##  mean of x  mean of y
## 0.06820063 0.07745286
```

```
young_per_country <- ages_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Young, type = 'mean_sd') %>%
  arrange(UserCountry3)
old_per_country <- ages_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Old, type = 'mean_sd') %>%
  arrange(UserCountry3)
man_woman_per_country <- ages_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Man_Woman, type = 'mean_sd') %>%
  arrange(UserCountry3)
ages_per_country <- full_join(young_per_country, old_per_country,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n')) %>%
  rename(mean_young = mean.x,
         sd_young = sd.x,
         mean_old = mean.y,
         sd_old = sd.y,)
ages_per_country <- full_join(ages_per_country, man_woman_per_country,
                                by = c('UserCountry3' = 'UserCountry3', 'n' = 'n')) %>%
        rename(mean_mid = mean,
         sd_mid = sd) %>%
  select(UserCountry3, n, mean_young, mean_mid, mean_old, sd_young, sd_mid, sd_old)
```

**Difference in age per Member State** When this is plotted, you can see that the young are only saved a little more in comparison to the ‘normal’ non-specified age. Whereas the elderly are drastically saved less than the non-elderly.

```r
## plots means per country for different ages
ggplot(ages_per_country, aes(x = UserCountry3)) +
  geom_point(aes(y = mean_young), color = 'light blue', size = 2.5) +
  geom_point(aes(y = mean_mid), color = '#CC9933', size = 2.5) +
  geom_point(aes(y = mean_old), color = 'dark blue', size = 2.5) +
  scale_x_discrete(name = "Member States") +
  scale_y_continuous(name = 'Mean per age class') +
  theme(legend.position = "right", text = element_text(size = 12),
        axis.text = element_text(size = 9)) +
  labs(title = 'Mean Young per Country') +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```




### Social Status

Finally, the same is done for social status. A lot is said about the ethical correctness of collecting this data. That is something I discuss in the literature review. For now, let us look at what the data has to say about it.

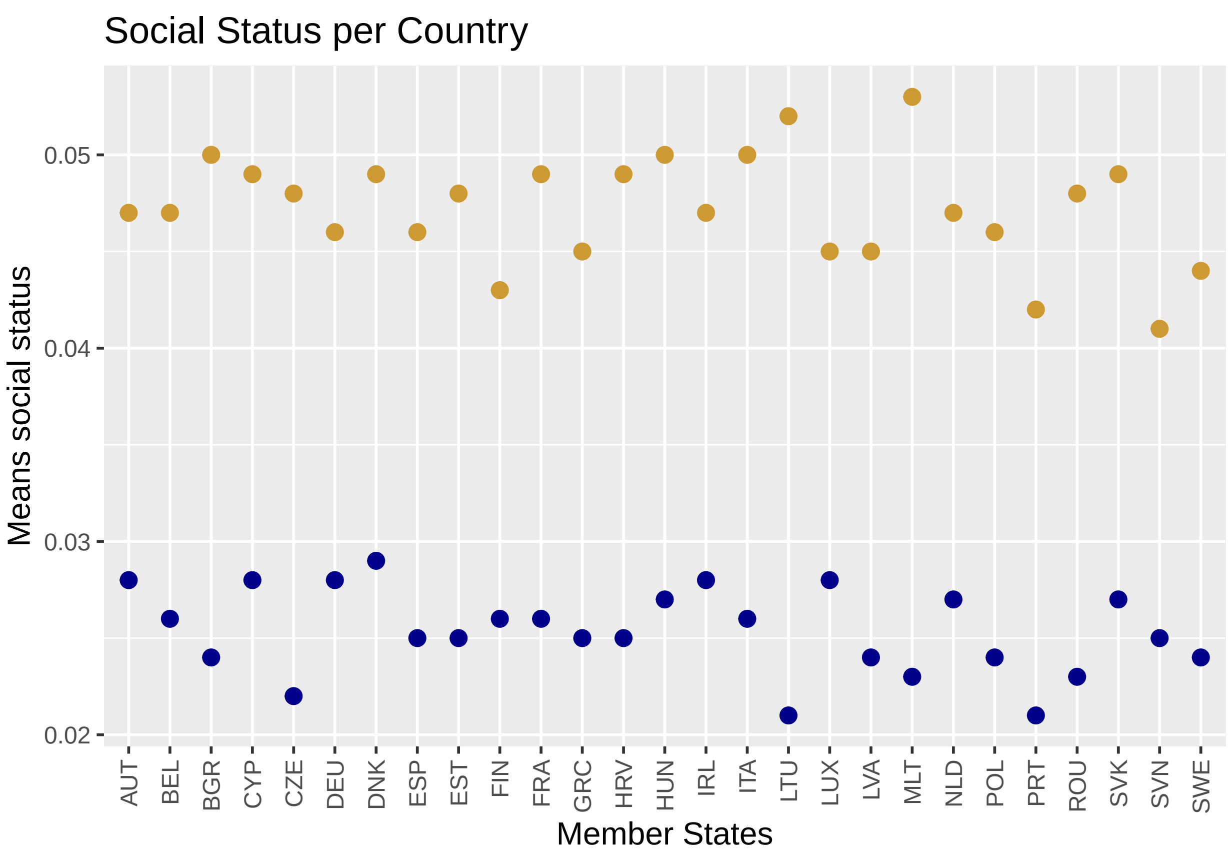
```r
### Social status
social_status_total <- Final_Frame %>%
  mutate(Low = (Homeless + Criminal)/2,
         High = (MaleExecutive + MaleAthlete + MaleDoctor +
           FemaleExecutive + FemaleAthlete + FemaleDoctor)/6) %>%
  summarise(ResponseID, UserCountry3, Low, High)
t.test(social_status_total$Low, social_status_total$High)
```

```
##
##  Welch Two Sample t-test
##
## data:  social_status_total$Low and social_status_total$High
## t = -136.73, df = 8674459, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02188739 -0.02126878
## sample estimates:
##  mean of x  mean of y
## 0.02562456 0.04720265
```

```
Low_per_country <- social_status_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Low, type = 'mean_sd') %>%
  arrange(UserCountry3)
High_per_country <- social_status_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(High, type = 'mean_sd') %>%
  arrange(UserCountry3)
Social_status_per_country <- full_join(Low_per_country, High_per_country,
                                       by = c('UserCountry3' = 'UserCountry3', 'n' = 'n')) %>%
  rename(mean_low = mean.x,
         sd_low = sd.x,
         mean_high = mean.y,
         sd_high = sd.y) %>%
  summarise(UserCountry3, n, mean_low, mean_high, sd_low, sd_high)
```

**Difference in social status per Member State** This plot shows the difference in social status for 'High' and 'Low' status High is defined as: Executives, Doctors and Athletes Low is defined as: Homeless and Criminal

```
## plot social status per country
ggplot(Social_status_per_country, aes(x = UserCountry3)) +
  geom_point(aes(y = mean_high), color = '#CC9933', size = 2.5) +
  geom_point(aes(y = mean_low), color = 'dark blue', size = 2.5) +
  scale_x_discrete(name = "Member States") +
  scale_y_continuous(name = 'Means social status') +
  theme(legend.position = "right", text = element_text(size = 12),
        axis.text = element_text(size = 9),
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) +
  labs(title = 'Social Status per Country')
```




**EDA Clustering**

For the exploratory analysis, two variables will be clustered to gain some insights into how they relate to each other and test the K-Means clustering algortihm for this data set. First, the data is manipulated to get the aggregates per country: #### Gender per country

```
## creates data set gender per country
gender_total <- Final_Frame %>%
  mutate(MenTotal = Man + OldMan + Boy + LargeMan + MaleExecutive + MaleAthlete + MaleDoctor,
         WomenTotal = Woman + OldWoman + Girl + LargeWoman + FemaleExecutive + FemaleAthlete + FemaleDo
  summarise(ResponseID, UserCountry3, Intervention, DefaultChoiceIsOmission, DiffNumberOFCharacters, Me

women_per_country <- gender_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(WomenTotal, type = 'mean_sd') %>%
  arrange(UserCountry3)
men_per_country <- gender_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(MenTotal, type = 'mean_sd') %>%
  arrange(UserCountry3)
gender_per_country <- full_join(men_per_country, women_per_country, by = c('UserCountry3' = 'UserCountr
  rename(mean_men = mean.x,
         sd_men = sd.x,
         mean_women = mean.y,
         sd_women = sd.y) %>%
  mutate(difference = mean_women -  mean_men) %>%
```

```
  arrange(desc(difference)) %>%
  select(UserCountry3, n, mean_men, mean_women, difference, sd_men, sd_women) %>%
  arrange(difference)
```

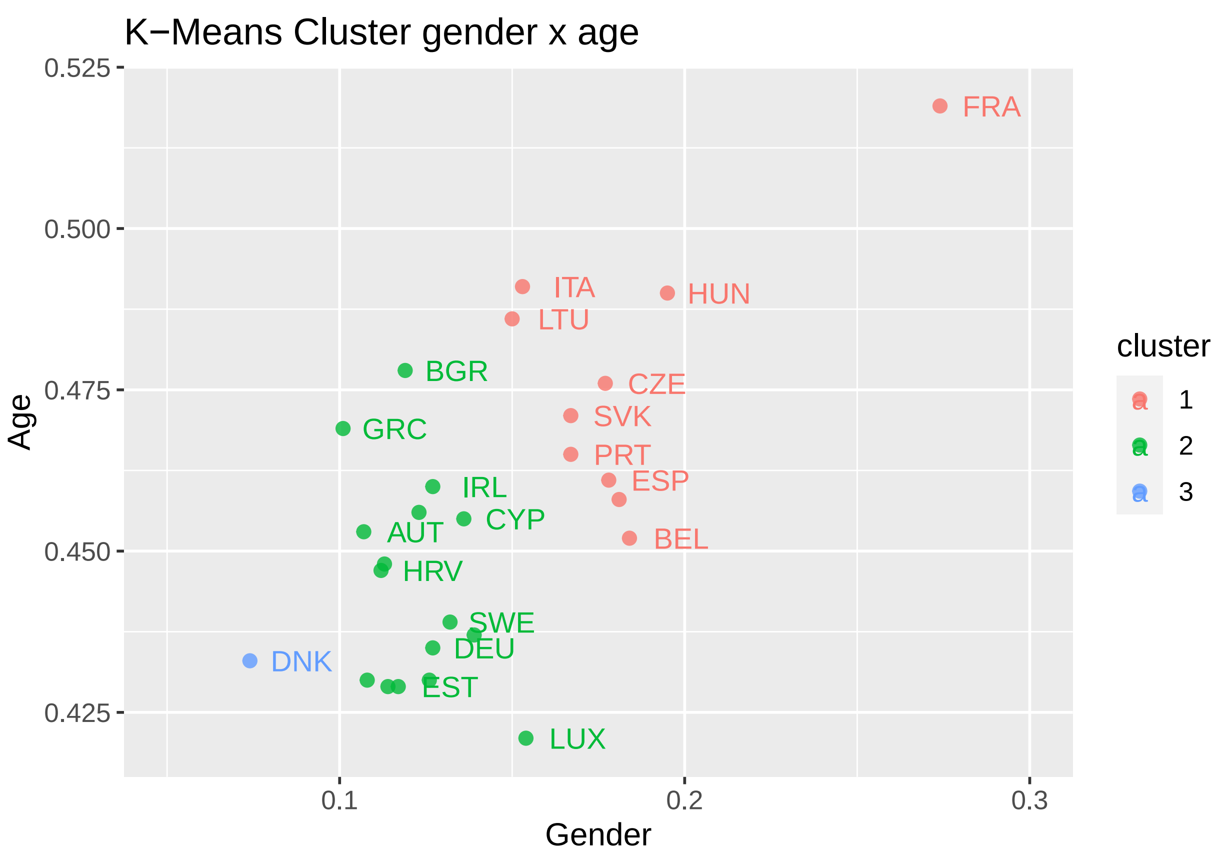
```
### create data set age per country
### combines all ages
ages_total <- Final_Frame %>%
  mutate(Young = Boy + Girl + Stroller + Pregnant,
         Man_Woman = Man + Woman,
         Old = OldMan + OldWoman) %>%
  summarise(ResponseID, UserCountry3, DiffNumberOFCharacters, Young, Old, Man_Woman)

young_per_country <- ages_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Young, type = 'mean_sd') %>%
  arrange(UserCountry3)
old_per_country <- ages_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Old, type = 'mean_sd') %>%
  arrange(UserCountry3)
man_woman_per_country <- ages_total %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Man_Woman, type = 'mean_sd') %>%
  arrange(UserCountry3)
ages_per_country <- full_join(young_per_country, old_per_country,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n')) %>%
  rename(mean_young = mean.x,
         sd_young = sd.x,
         mean_old = mean.y,
         sd_old = sd.y,)
ages_per_country <- full_join(ages_per_country, man_woman_per_country,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n')) %>%
  rename(mean_mid = mean,
         sd_mid = sd) %>%
  select(UserCountry3, n, mean_young, mean_mid, mean_old, sd_young, sd_mid, sd_old)
```

```
### Create Data set for clustering gender x age
age_gender <- full_join(ages_per_country, gender_per_country,
                        by = c('UserCountry3' = 'UserCountry3', 'n' = 'n')) %>%
                        mutate(difference_age = mean_young - mean_old) %>%
                        summarise(UserCountry3, mean_men, mean_women, difference, mean_young, mean_old,
```

**Age per country**

**Cluster gender x age** The plot below shows again how France is a real outlier in this data set. The Member States are divided in three clusters in this plot, this is chosen based on visual clusters.

```r
age_gender %>% select(-UserCountry3) %>% kmeans(centers=3) -> km

ag_clustered <- data.frame(age_gender, cluster = factor(km$cluster))

ggplot(ag_clustered, aes(x = difference, y = difference_age, color = cluster)) +
  geom_point(size = 2, alpha=0.8) +
  geom_text(label = ag_clustered$UserCountry3, nudge_x=0.015, check_overlap = T) +
  theme(legend.position = "right", text = element_text(size = 12),
        axis.text = element_text(size = 10)) +
  labs(title = 'K-Means Cluster gender x age',
       x = 'Gender',
       y = 'Age') +
  xlim(0.05,0.3) +
  ylim(0.42,0.52)
```




```r
ag_clustered %>% summarise(UserCountry3, cluster)
```

```
##    UserCountry3 cluster
## 1           AUT       2
## 2           BEL       1
## 3           BGR       2
## 4           CYP       2
## 5           CZE       1
## 6           DEU       2
## 7           DNK       3
## 8           ESP       1
```

```
## 9           EST       2
## 10          FIN       2
## 11          FRA       1
## 12          GRC       2
## 13          HRV       2
## 14          HUN       1
## 15          IRL       2
## 16          ITA       1
## 17          LTU       1
## 18          LUX       2
## 19          LVA       2
## 20          MLT       1
## 21          NLD       2
## 22          POL       2
## 23          PRT       1
## 24          ROU       2
## 25          SVK       1
## 26          SVN       2
## 27          SWE       2
```

**Some more data manipulation** Below, the code is used to get the aggregated values of every variable in the data set, per country.

```
Country_Intervention <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Intervention, type = 'mean') %>%
  rename(Intervention = mean)
Country_NumChar <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(DiffNumberOFCharacters, type = 'mean')%>%
  rename(DiffNumberOFCharacters = mean)
Country_Man <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Man, type = 'mean')%>%
  rename(Man = mean)
Country_Woman <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Woman, type = 'mean')%>%
  rename(Woman = mean)
Country_Pregnant <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Pregnant, type = 'mean')%>%
  rename(Pregnant = mean)
Country_Stroller <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Stroller, type = 'mean')%>%
  rename(Stroller = mean)
Country_OldMan <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(OldMan, type = 'mean')%>%
  rename(OldMan = mean)
Country_OldWoman <- Final_Frame %>%
  group_by(UserCountry3) %>%
```

```
  get_summary_stats(OldWoman, type = 'mean')%>%
  rename(OldWoman = mean)
Country_Boy <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Boy, type = 'mean')%>%
  rename(Boy = mean)
Country_Girl <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Girl, type = 'mean')%>%
  rename(Girl = mean)
Country_Homeless <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Homeless, type = 'mean')%>%
  rename(Homeless = mean)
Country_LargeWoman <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(LargeWoman, type = 'mean')%>%
  rename(LargeWoman = mean)
Country_LargeMan <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(LargeMan, type = 'mean')%>%
  rename(LargeMan = mean)
Country_Criminal <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Criminal, type = 'mean')%>%
  rename(Criminal = mean)
Country_MaleAth <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(MaleAthlete, type = 'mean')%>%
  rename(MaleAthlete = mean)
Country_MaleExec <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(MaleExecutive, type = 'mean')%>%
  rename(MaleExecutive = mean)
Country_MaleDoctor <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(MaleDoctor, type = 'mean')%>%
  rename(MaleDoctor = mean)
Country_FemaleExec <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(FemaleExecutive, type = 'mean')%>%
  rename(FemaleExecutive = mean)
Country_FemaleAth <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(FemaleAthlete, type = 'mean')%>%
  rename(FemaleAthlete = mean)
Country_FemaleDoctor <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(FemaleDoctor, type = 'mean')%>%
  rename(FemaleDoctor = mean)
Country_Cat <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Cat, type = 'mean')%>%
```

```
  rename(Cat = mean)
CountryDog <- Final_Frame %>%
  group_by(UserCountry3) %>%
  get_summary_stats(Dog, type = 'mean')%>%
  rename(Dog = mean)

Country_Averaged <- full_join(Country_Intervention, Country_NumChar,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                              keep = FALSE) %>%
  select(-variable.x, -variable.y)
# add Man
Country_Averaged <- full_join(Country_Averaged, Country_Man,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                              keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add Woman
Country_Averaged <- full_join(Country_Averaged, Country_Woman,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                              keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add Pregnant
Country_Averaged <- full_join(Country_Averaged, Country_Pregnant,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                              keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add Stroller
Country_Averaged <- full_join(Country_Averaged, Country_Stroller,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                              keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add OldMan
Country_Averaged <- full_join(Country_Averaged, Country_OldMan,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                              keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add OldWoman
Country_Averaged <- full_join(Country_Averaged, Country_OldWoman,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                              keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add Boy
Country_Averaged <- full_join(Country_Averaged, Country_Boy,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                              keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add Girl
Country_Averaged <- full_join(Country_Averaged, Country_Girl,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                              keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add Homeless
Country_Averaged <- full_join(Country_Averaged, Country_Homeless,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
```

```r
                                keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add LargeWoman
Country_Averaged <- full_join(Country_Averaged, Country_LargeWoman,
                                by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                                keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add LargeMan
Country_Averaged <- full_join(Country_Averaged, Country_LargeMan,
                                by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                                keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add Criminal
Country_Averaged <- full_join(Country_Averaged, Country_Criminal,
                                by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                                keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add MaleExecutive
Country_Averaged <- full_join(Country_Averaged, Country_MaleExec,
                                by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                                keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add MaleAthlete
Country_Averaged <- full_join(Country_Averaged, Country_MaleAth,
                                by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                                keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add MaleDoctor
Country_Averaged <- full_join(Country_Averaged, Country_MaleDoctor,
                                by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                                keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add FemaleExecutive
Country_Averaged <- full_join(Country_Averaged, Country_FemaleExec,
                                by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                                keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add FemaleAthlete
Country_Averaged <- full_join(Country_Averaged, Country_FemaleAth,
                                by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                                keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add FemaleDoctor
Country_Averaged <- full_join(Country_Averaged, Country_FemaleDoctor,
                                by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                                keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add Dog
Country_Averaged <- full_join(Country_Averaged, CountryDog,
                                by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                                keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)
# add Cat
```

```r
Country_Averaged <- full_join(Country_Averaged, Country_Cat,
                              by = c('UserCountry3' = 'UserCountry3', 'n' = 'n'),
                              keep = FALSE)
Country_Averaged <- Country_Averaged %>% select(-variable)

#write.csv(Country_Averaged, 'MM_EU//Country_Averaged.csv')
```

## Analysis

In the following part, the analysis conducted to answer the research question is done. This starts with clustering the data, finding the optimal number of clusters for this data. Clustering is done with Wards method (Ward.D2 in the stats package). To get robust outcomes, two different distances are used to check whether they influence the outcome: Euclidean and Manhattan distance.

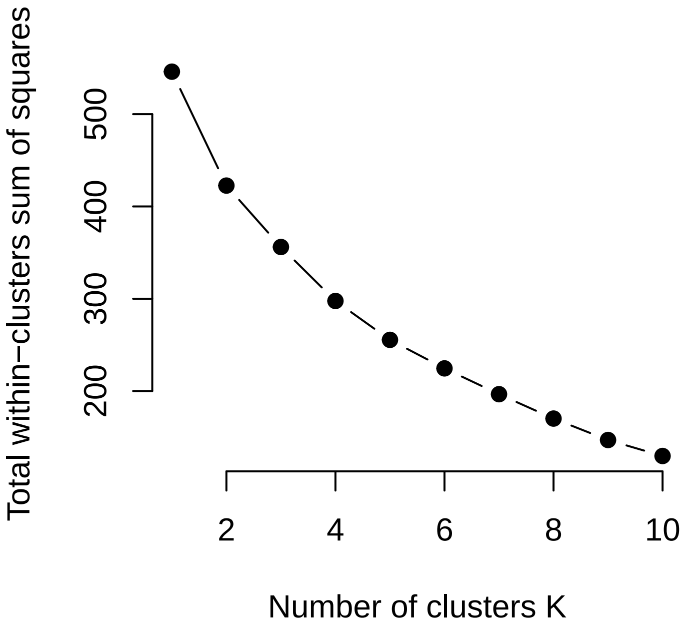
```r
# First cluster: Ward.D2, Euclidean distance on 20 variables (intervention + all attributes, diffnumcha
Country_Averaged <- read.csv('MM_EU/Country_Averaged.csv', header=TRUE, sep=',')
Country_Averaged <- Country_Averaged %>% select(-X, -DiffNumberOFCharacters)
distances_Eucl <- dist(Country_Averaged[3:23], method = 'euclidean')
Cluster_D2_Eucl <- hclust(distances_Eucl, method = 'ward.D2')
Labels_Order_D2_Eucl <- Country_Averaged$UserCountry3[Cluster_D2_Eucl$order]
scaled_data = as.matrix(scale(Country_Averaged[3:23]))
kmm = kmeans(scaled_data, 9, nstart=50, iter.max = 15)
kmm
```

```
## K-means clustering with 9 clusters of sizes 5, 3, 1, 6, 4, 5, 1, 1, 1
##
## Cluster means:
##   Intervention        Man      Woman   Pregnant   Stroller     OldMan
## 1    0.2917535 -0.5738993 -0.2203317 -1.1146274 -0.5766414  0.0459618
## 2   -1.3247988 -0.2454175 -0.3510780  0.3606147  0.6897624 -0.8529450
## 3   -1.3694550 -0.3020523  2.9175790  1.5408084  0.6897624 -3.0803247
## 4    0.6177433  0.7740089 -0.3783168  0.4343768 -0.3034955  0.6187166
## 5   -0.5321523  0.1227087  1.1606759  1.0982358  0.6897624 -0.5745225
## 6    0.5061029  0.1396992 -0.1549586 -0.5835402 -0.5021471  0.4993927
## 7    0.9080082  2.0766093 -1.4951079 -1.9997726 -1.5450679  0.6187166
## 8   -1.6373918 -1.1515743  0.4660863  1.0982358 -0.4276527 -0.3358747
## 9    0.5061029 -2.8506183 -1.3316751 -0.2294821  3.6695361  1.2153361
##       OldWoman        Boy       Girl    Homeless  LargeWoman   LargeMan
## 1  0.07972231 -0.8142237 -0.1505205  0.08868916  0.53615464  0.5910739
## 2 -1.43650570  0.4118641  0.7526024 -1.81669722 -0.72130974 -1.4380767
## 3 -0.89499570  0.9821375  3.0104096  0.62940691  1.88562861 -1.0541833
## 4  0.42493493  0.4118641 -0.6197902  0.75814923  0.07099505  1.1394930
## 5  0.01880243  0.7682850  0.9186176  0.33973668  0.88885806 -0.0670290
## 6  0.03910906 -0.5148301 -0.6286444 -0.83825557 -0.69063988 -0.3631753
## 7  0.93260056 -1.7979452 -1.1067682  2.17431478  0.65883409 -0.5606062
## 8  0.32340181 -0.9425352  0.3541658  0.24317994 -1.02800837 -1.5477605
## 9  0.72953431  1.6236950 -0.5755195  0.24317994 -2.56150152 -0.2315547
##      Criminal MaleExecutive MaleAthlete MaleDoctor FemaleExecutive
## 1 -0.84200953   -1.06527312 -0.88048907 -0.7143766     -0.44081436
## 2 -0.59354770   -0.05885487  1.16988059  0.1911712      0.08477199
## 3 -0.09662404   -1.64793632 -1.04673526 -1.5293696      2.96701975
## 4  0.95933873    0.77982701 -0.21550432  0.9608868     -0.33908797
```

```
## 5  0.55558826    0.07356859 -0.07696583  0.1006164      1.18680790
## 6 -0.32023969    0.31193080  0.80968051  0.4266136     -0.64426715
## 7 -0.09662404    1.00053277 -2.15504319 -1.8010340      0.67817594
## 8 -1.58739502    1.00053277  1.72403455 -2.0726983     -1.61066786
## 9  1.39414693   -1.38308941 -0.07696583  0.1006164      0.42385996
##   FemaleAthlete FemaleDoctor        Dog         Cat
## 1   -1.01177664   -0.4975227  1.1961624  1.07379648
## 2    1.02650678   -0.1049967  0.5069686  0.33380453
## 3    1.27507793    2.5118435 -0.4502449 -0.33029080
## 4   -0.58920569   -0.5047917 -0.6257340 -0.63071488
## 5    0.37400752    0.8218009 -1.0963640 -1.35015150
## 6   -0.19149185   -0.8028207  0.2772374  0.58046852
## 7   -0.09206339    0.9853534 -0.1630808 -0.23542004
## 8    2.64221925    2.0757035  0.7941327  0.52354606
## 9    1.15079236    0.9853534 -0.9288516 -0.04567852
##
## Clustering vector:
##  [1] 4 5 2 9 2 4 4 1 6 1 3 4 6 5 4 5 2 7 6 8 4 6 1 6 5 1 1
##
## Within cluster sum of squares by cluster:
## [1] 38.97287 15.46198  0.00000 40.24414 20.08194 32.16061  0.00000  0.00000
## [9]  0.00000
##  (between_SS / total_SS =  73.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

**Elbow Method** Elbow method is traditionally used to find the optimal k for the data: This is done to prevent over- and underfitting.

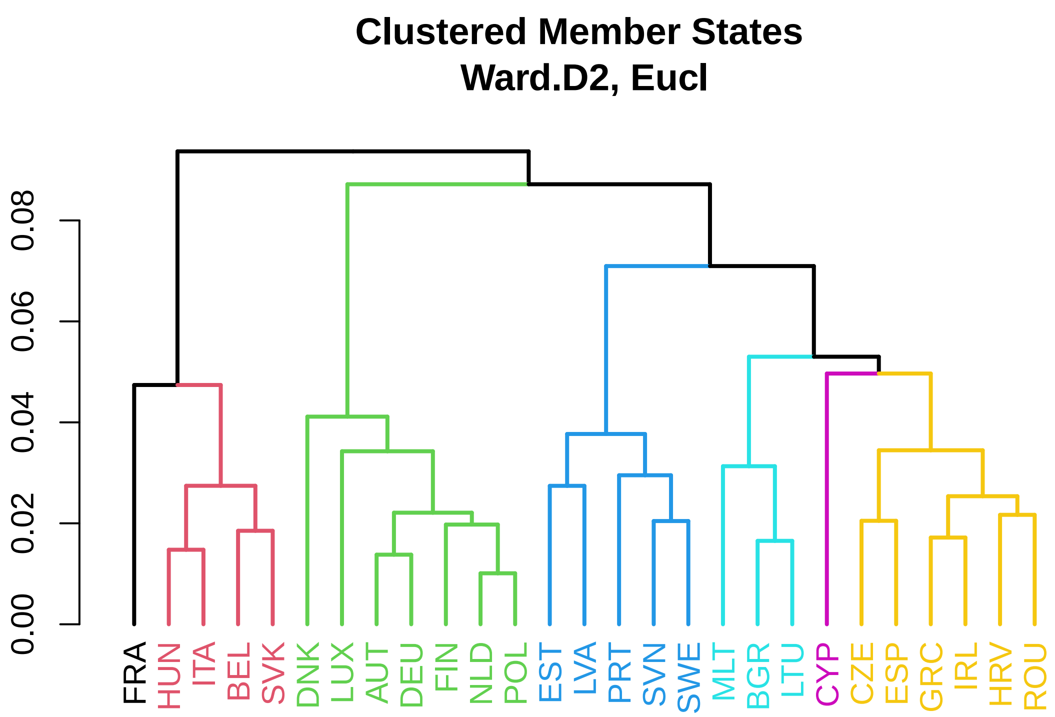
```r
# Elbow method for k = 2 - 15
k.max <- 10
data <- scaled_data
wss <- sapply(1:k.max,
              function(k){kmeans(data, k, nstart=50,iter.max = 10 )$tot.withinss})
plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```




Based on the elbow method, K = 7 was chosen as the optimal number of clusters.

#### Dendrogram 1: Euclidean distance, k = 7, Ward's method

```
dendro1 <- as.dendrogram(Cluster_D2_Eucl)
Labels_Order_D2_Eucl <- Country_Averaged$UserCountry3[Cluster_D2_Eucl$order]
labels(dendro1) <- Labels_Order_D2_Eucl
plotje <- set(dendro1, "labels_cex") %>%
  set("labels_col", value = c(1:7), k=7) %>%
  set("branches_lwd", 2) %>%
  set("branches_k_color", value = 1:7, k = 7) %>%
  plot(main = "Clustered Member States \nWard.D2, Eucl")
```

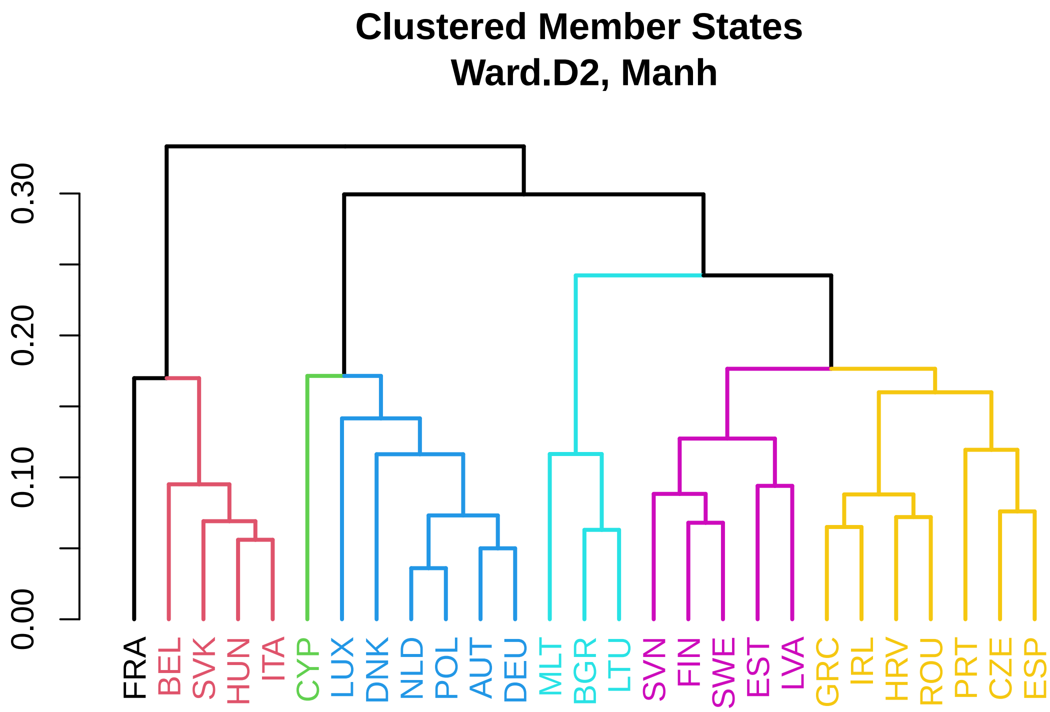



```
dendro1 %>% get_nodes_attr('height') ## get node's height for every split
```

```
##  [1] 0.09367896 0.04739726 0.00000000 0.02743173 0.01476482 0.00000000
##  [7] 0.00000000 0.01852026 0.00000000 0.00000000 0.08716034 0.04112872
## [13] 0.00000000 0.03427244 0.00000000 0.02209977 0.01378405 0.00000000
## [19] 0.00000000 0.01974842 0.00000000 0.01009950 0.00000000 0.00000000
## [25] 0.07095257 0.03769704 0.02742262 0.00000000 0.00000000 0.02951836
## [31] 0.00000000 0.02044505 0.00000000 0.00000000 0.05300009 0.03131028
## [37] 0.00000000 0.01652271 0.00000000 0.00000000 0.04966411 0.00000000
## [43] 0.03446496 0.02049390 0.00000000 0.00000000 0.02536730 0.01717556
## [49] 0.00000000 0.00000000 0.02167948 0.00000000 0.00000000
```

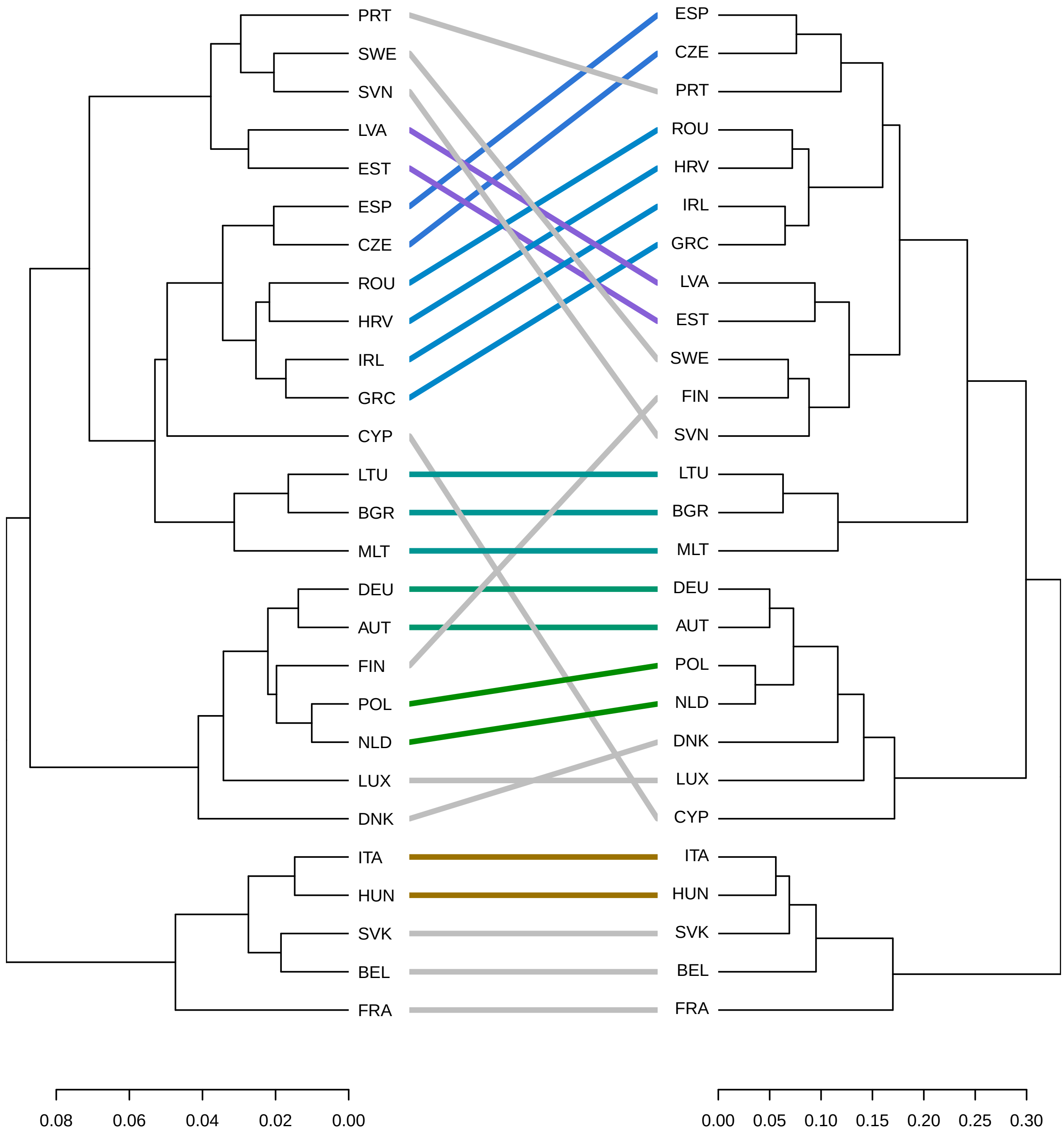
**Dendrogram 2: Manhattan distance, k = 7, Ward's method**

```
distances_Manh <- dist(Country_Averaged[3:23], method='manhattan', labels(Country_Averaged$UserCountry3)
cluster_d2_manh <- hclust(distances_Manh, method = 'ward.D2')
dendro2 <- as.dendrogram(cluster_d2_manh)
Labels_Order_D2_manh <- Country_Averaged$UserCountry3[cluster_d2_manh$order]
labels(dendro2) <- Labels_Order_D2_manh
plotje_manh <- set(dendro2, "labels_cex") %>%
  set("labels_col", value = c(1:7), k=7) %>%
  set("branches_lwd", 2) %>%
  set("branches_k_color", value = 1:7, k = 7) %>%
  plot(main = "Clustered Member States \nWard.D2, Manh")
```




**Comparing the two dendrograms** As can be seen in the plots, the clusters themselves are quite stable. The only difference is that the place where they split off are different. In other words, the between clusters are stable, the within clusters differ a bit.

```
## Compare the two dendrograms using dendextend
dendlist(dendro1, dendro2) %>%
  untangle(method='step1side') %>%
  tanglegram( ## plots two dendrograms to visually compare them
    highlight_distinct_edges = FALSE,
    highlight_branches_lwd = FALSE)
```




```
dendlist(dendro1, dendro2) %>%
  untangle(method='step1side') %>%
  entanglement() ## produces alignment quality, the lower the better
```

```
## [1] 0.1174483
```

```
## baker gamma is a correlation coefficient.
cor_bakers_gamma(dendro1, dendro2)
```

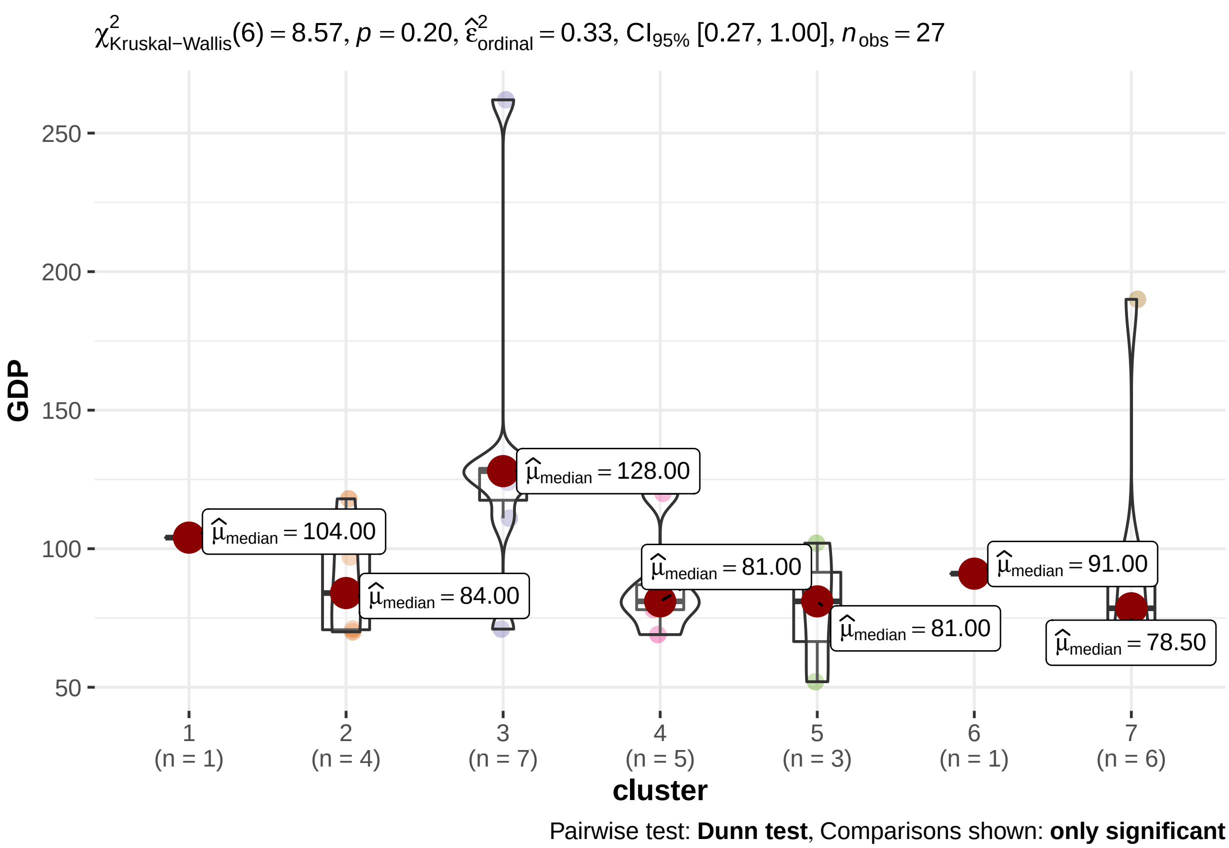
```
## [1] 0.8924898
```

Append clusters to data frame

```
EU_data <- read.csv('MM_EU/EU_data.csv', header = TRUE, sep = ',')
EU_info2 <- read.csv('EU_info2.csv', header = TRUE, sep = ';')

EU_data <- full_join(EU_data, EU_info2, by = "UserCountry3")
```

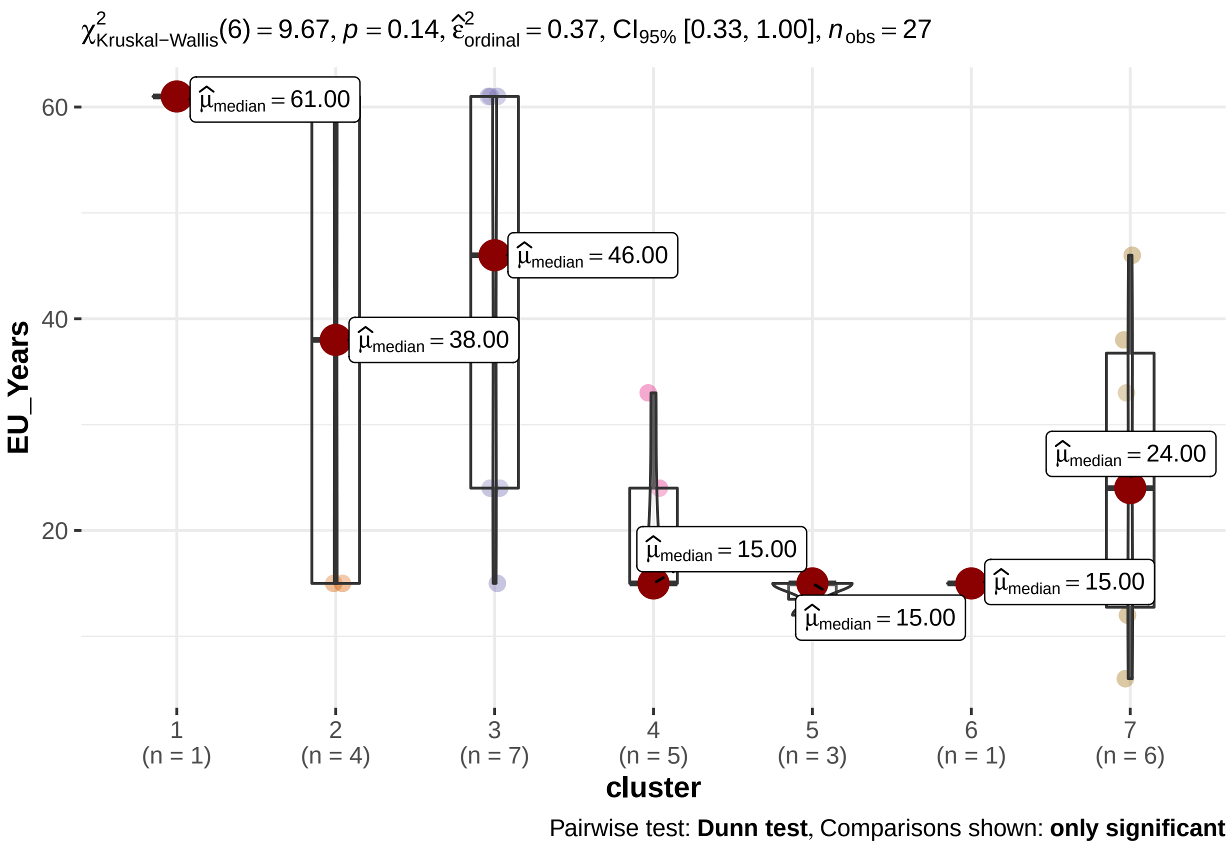
### Analysis: explain clusters

```
## Anova clusters on GDP
ggbetweenstats(
  data = EU_data,
  x = cluster,
  y = GDP,
  type = 'nonparametric',
  var.equal = FALSE,
  pairwise.display = 's'
)
```



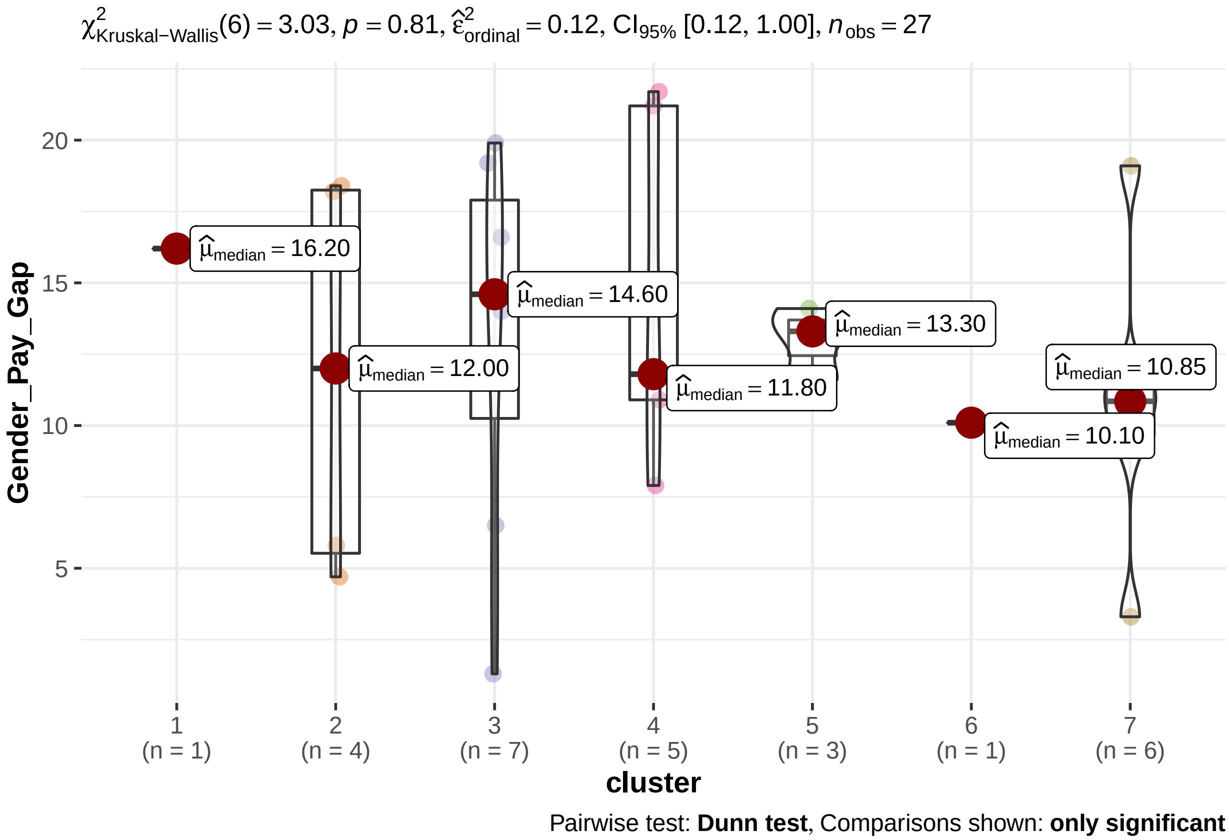

```
## Anova clusters on EU membership years
ggbetweenstats(
  data = EU_data,
  x = cluster,
  y = EU_Years,
```

```r
  type = 'nonparametric',
  var.equal = FALSE
)
```



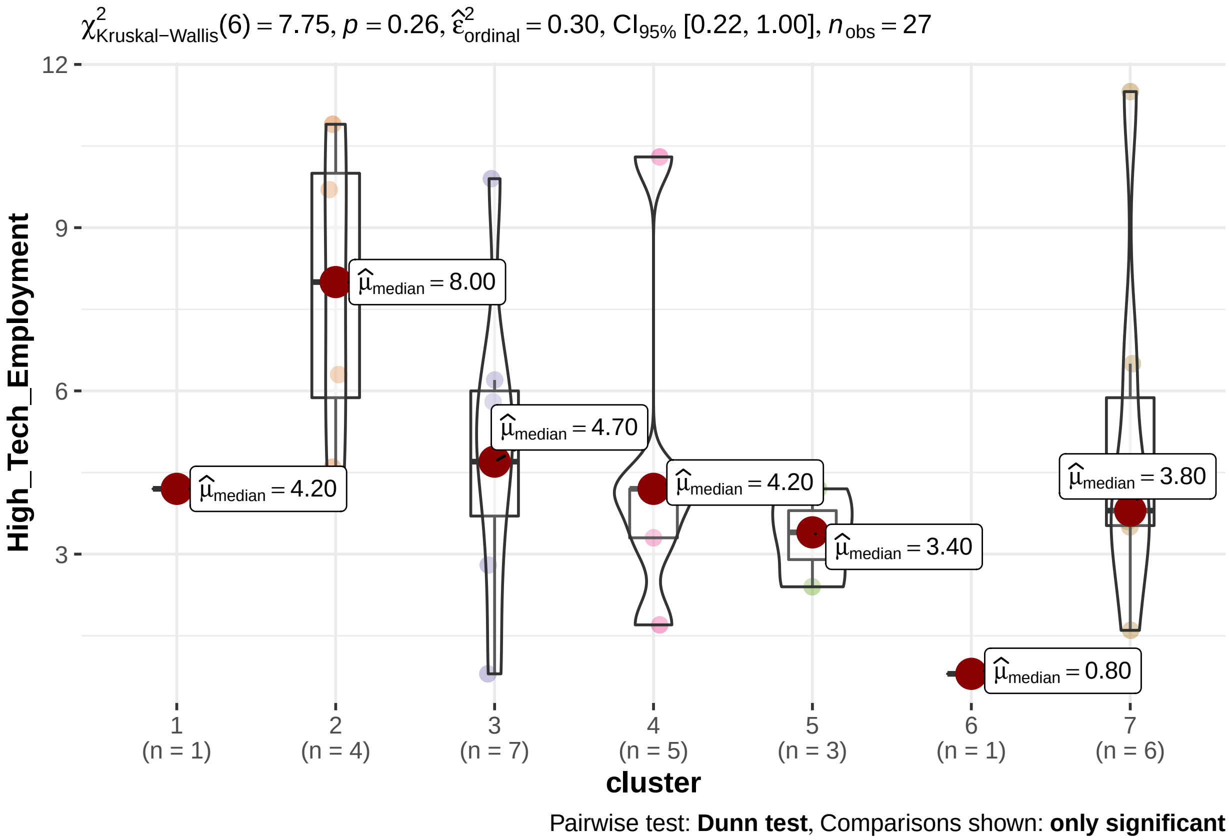

```r
## Anova clusters on Gender Pay Gap
ggbetweenstats(
  data = EU_data,
  x = cluster,
  y = Gender_Pay_Gap,
  type = 'nonparametric',
  var.equal = FALSE
)
```

```
## Warning: Groups with fewer than two data points have been dropped.

## Warning: Groups with fewer than two data points have been dropped.
```



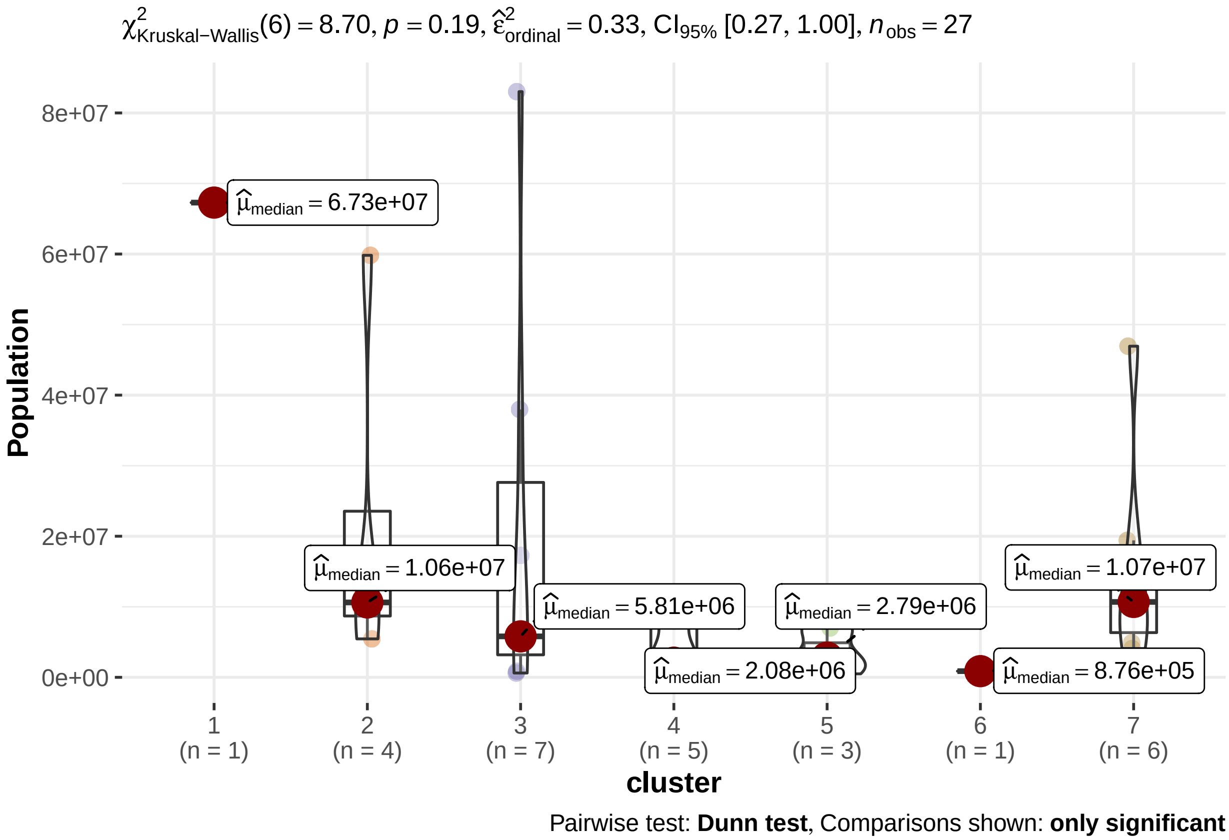

```
## Anova clusters on High Tech Employment
ggbetweenstats(
  data = EU_data,
  x = cluster,
  y = High_Tech_Employment,
  type = 'nonparametric',
  var.equal = FALSE
)
```

```
## Warning: Groups with fewer than two data points have been dropped.

## Warning: Groups with fewer than two data points have been dropped.
```



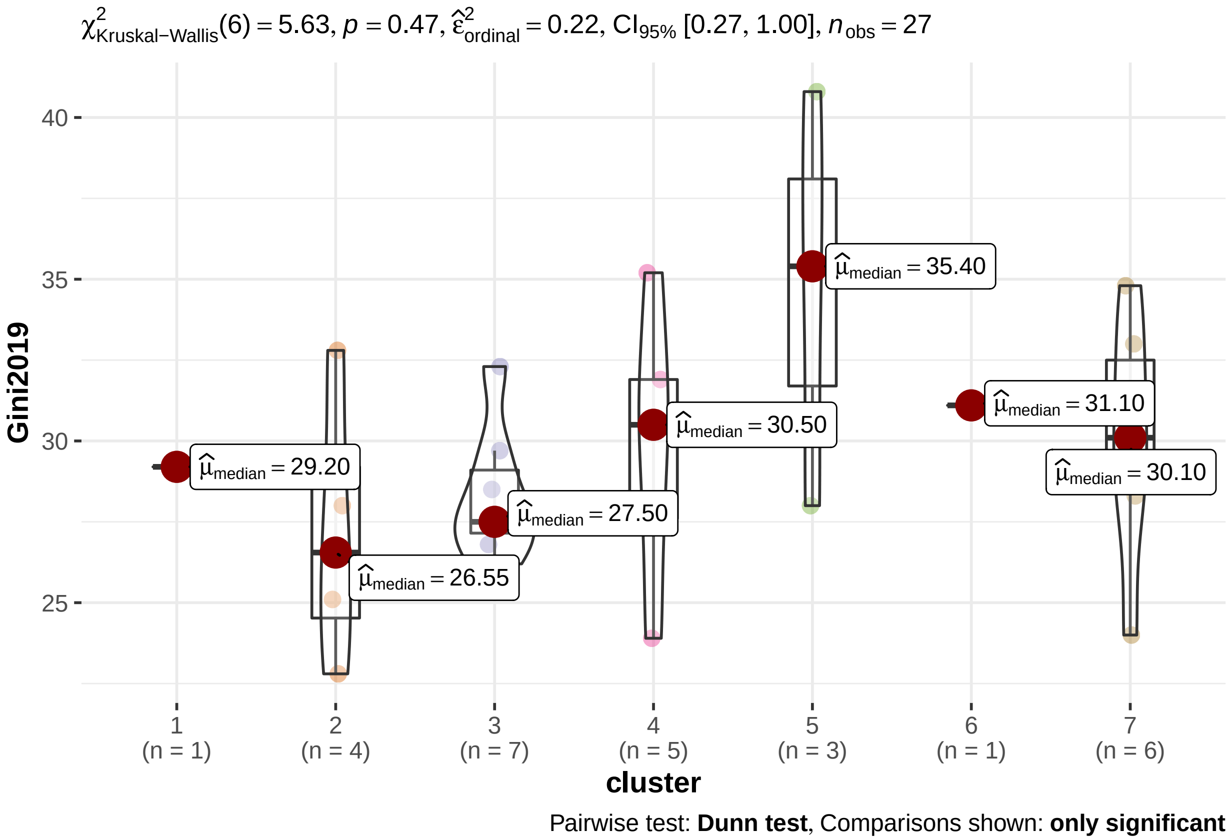

```r
## Anova clusters on Population
ggbetweenstats(
  data = EU_data,
  x = cluster,
  y = Population,
  type = 'nonparametric',
  var.equal = FALSE
)
```

```
## Warning: Groups with fewer than two data points have been dropped.

## Warning: Groups with fewer than two data points have been dropped.
```



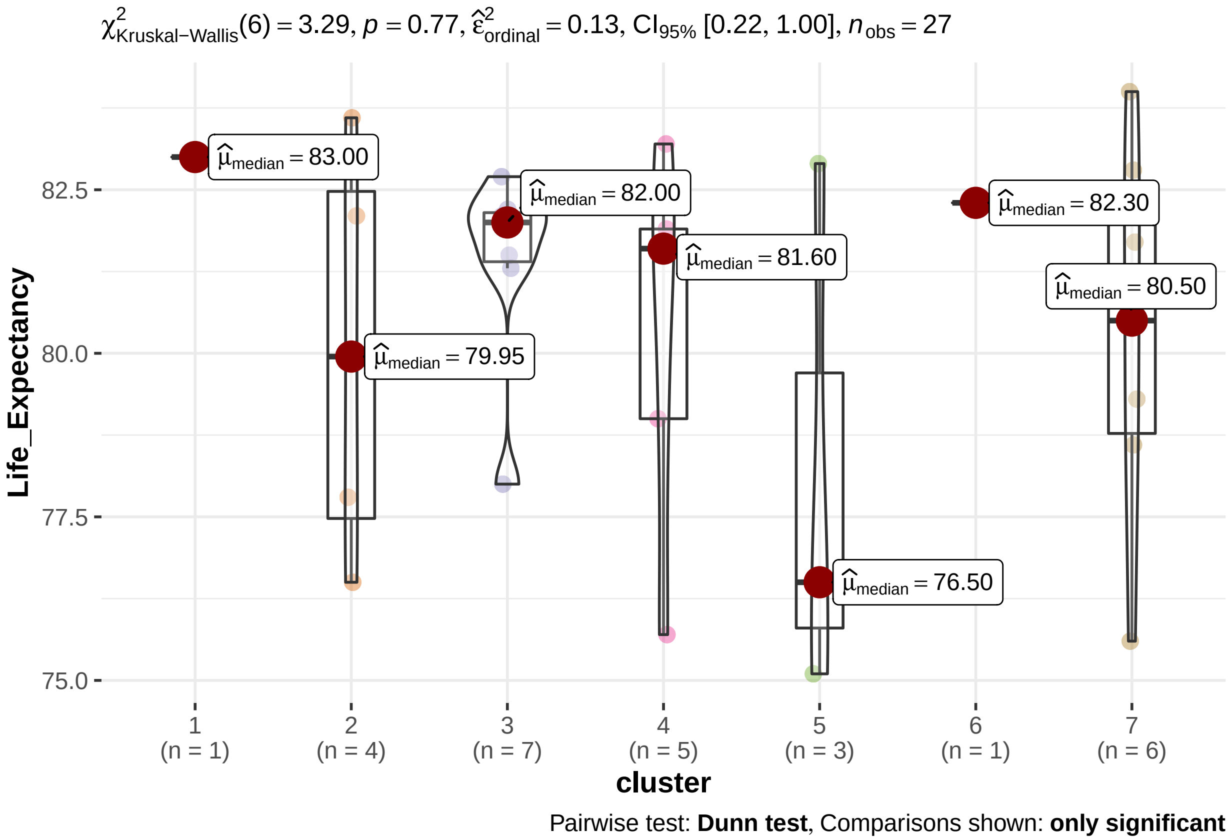

```
## ANOVA Gini2019
ggbetweenstats(
  data = EU_data,
  x = cluster,
  y = Gini2019,
  type = 'nonparametric',
  var.equal = FALSE
)
```

```
## Warning: Groups with fewer than two data points have been dropped.

## Warning: Groups with fewer than two data points have been dropped.
```

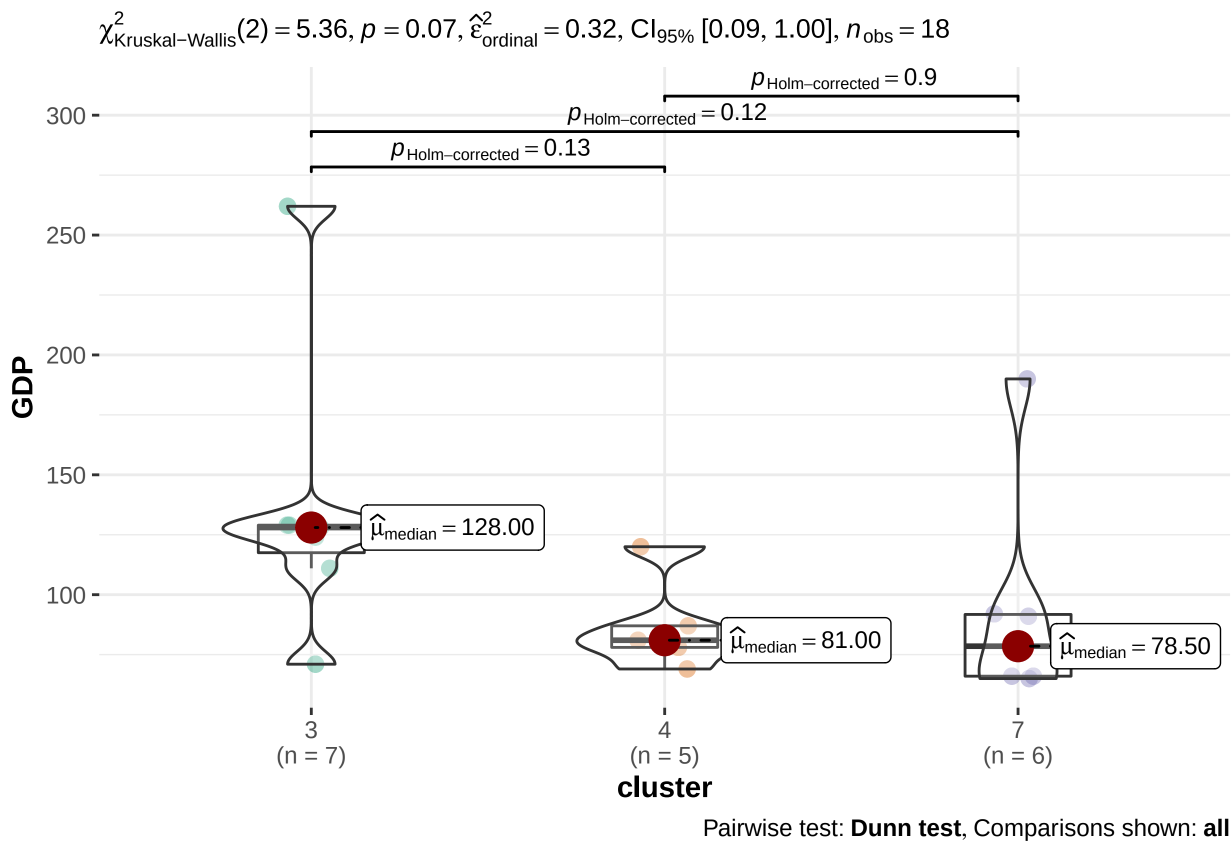



```
## ANOVA Life_Expectancy
ggbetweenstats(
  data = EU_data,
  x = cluster,
  y = Life_Expectancy,
  type = 'nonparametric',
  var.equal = FALSE
)
```

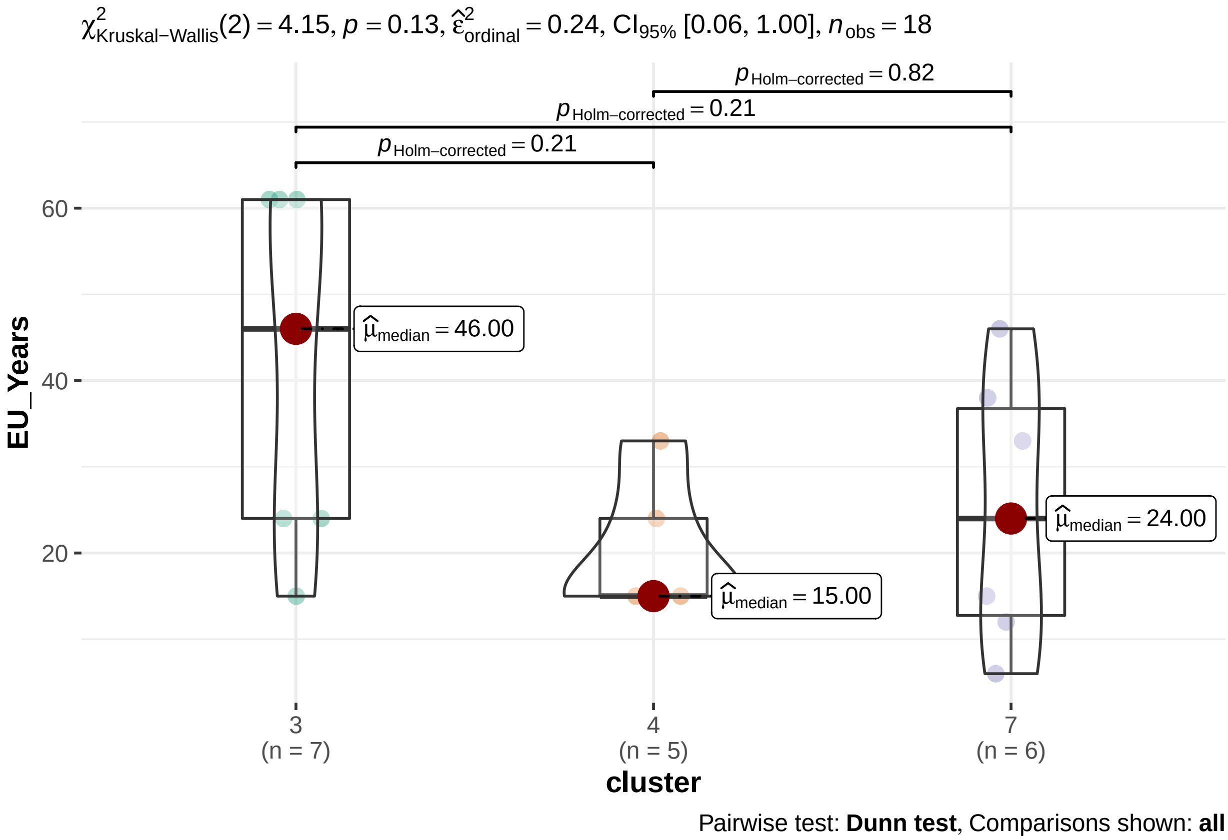
```
## Warning: Groups with fewer than two data points have been dropped.

## Warning: Groups with fewer than two data points have been dropped.
```



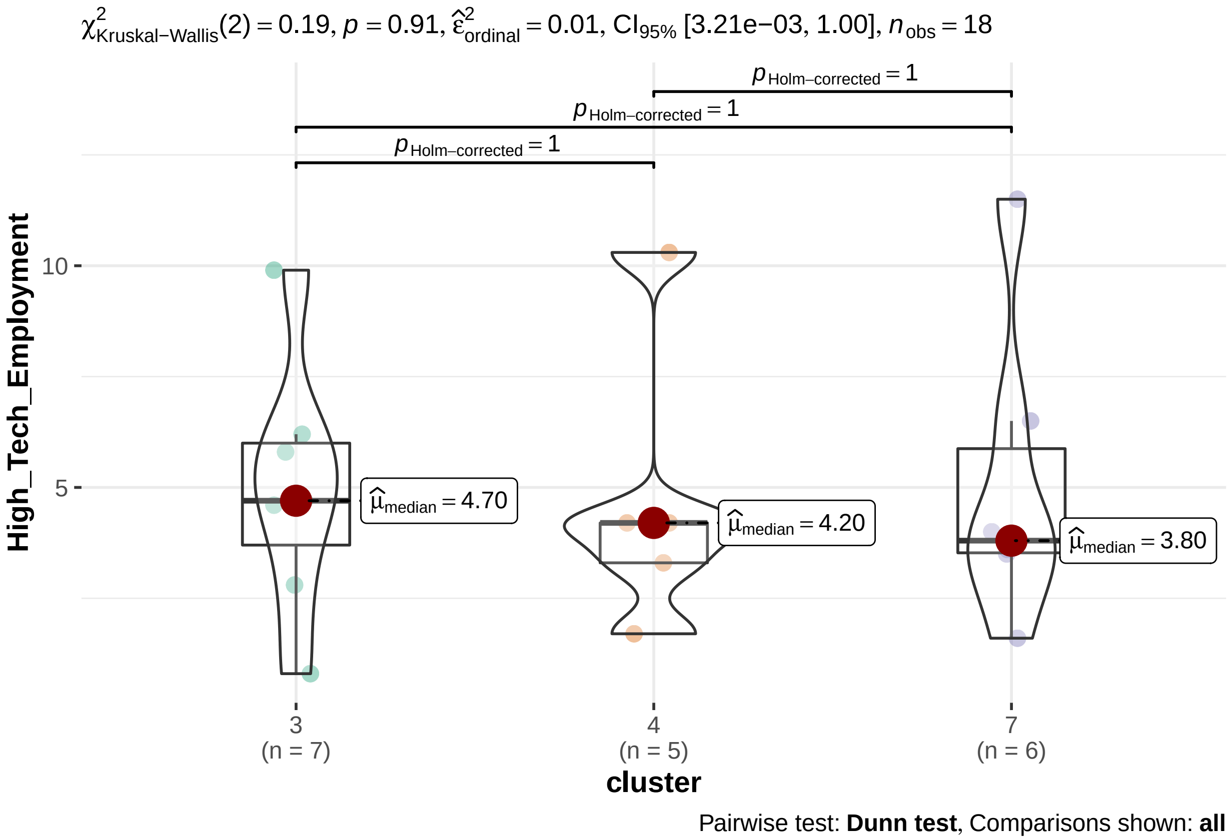

```r
big_clusters <- EU_data %>%
  filter(cluster1 == 0 &
           cluster2 == 0 &
           cluster5 == 0 &
           cluster6 == 0)

## anova with 3 biggest clusters on GDP
ggbetweenstats(
  data = big_clusters,
  x = cluster,
  y = GDP,
  type = 'nonparametric',
  var.equal = FALSE,
  pairwise.display = 'all'
)
```



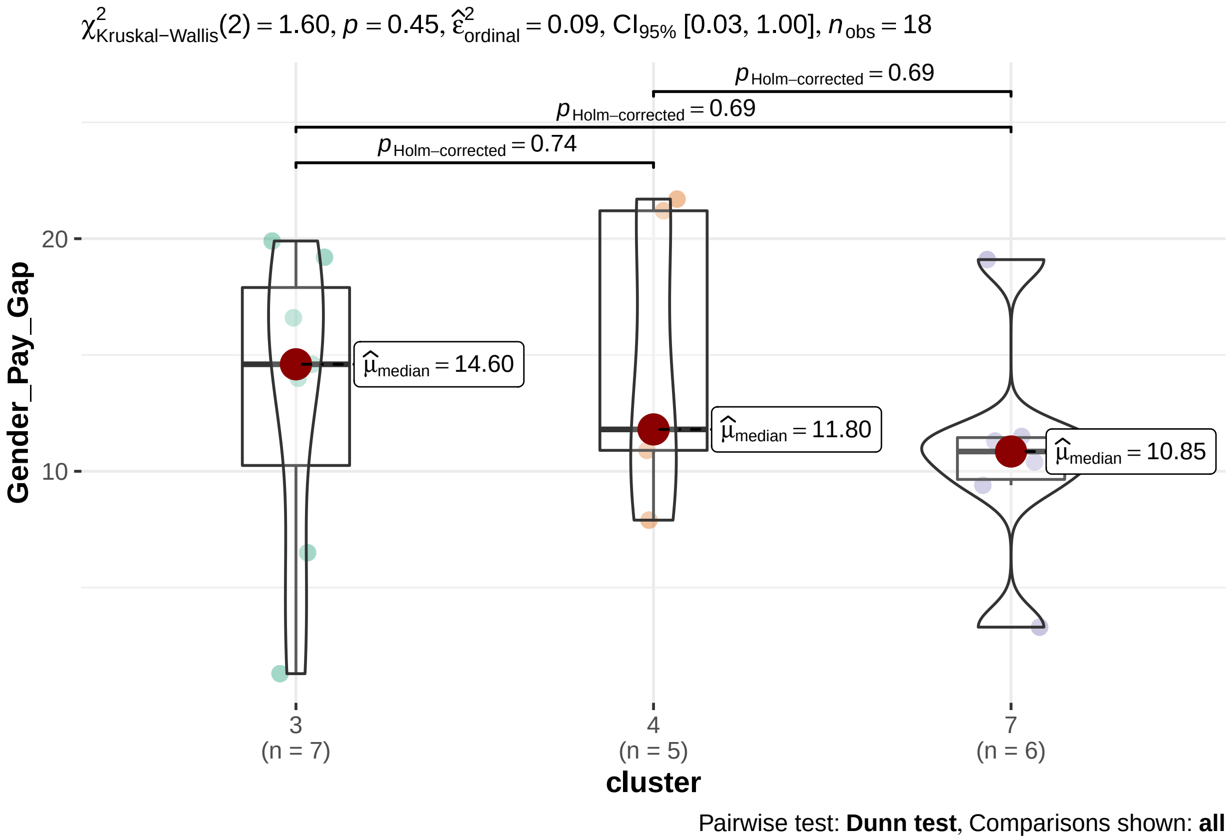

```r
## EU Years
ggbetweenstats(
  data = big_clusters,
  x = cluster,
  y = EU_Years,
  type = 'nonparametric',
  var.equal = FALSE,
  pairwise.display = 'all'
)
```



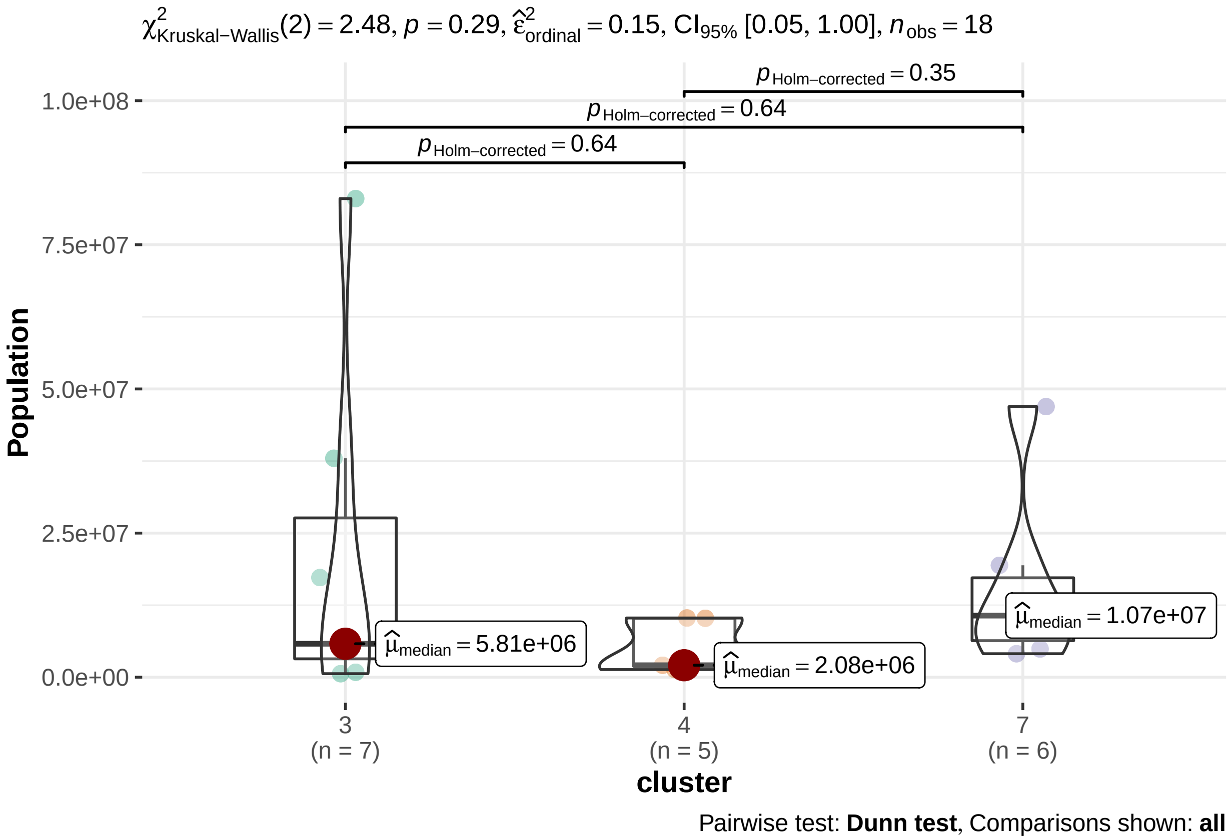

```r
## High_Tech_Employment
ggbetweenstats(
  data = big_clusters,
  x = cluster,
  y = High_Tech_Employment,
  type = 'nonparametric',
  var.equal = FALSE,
  pairwise.display = 'all'
)
```



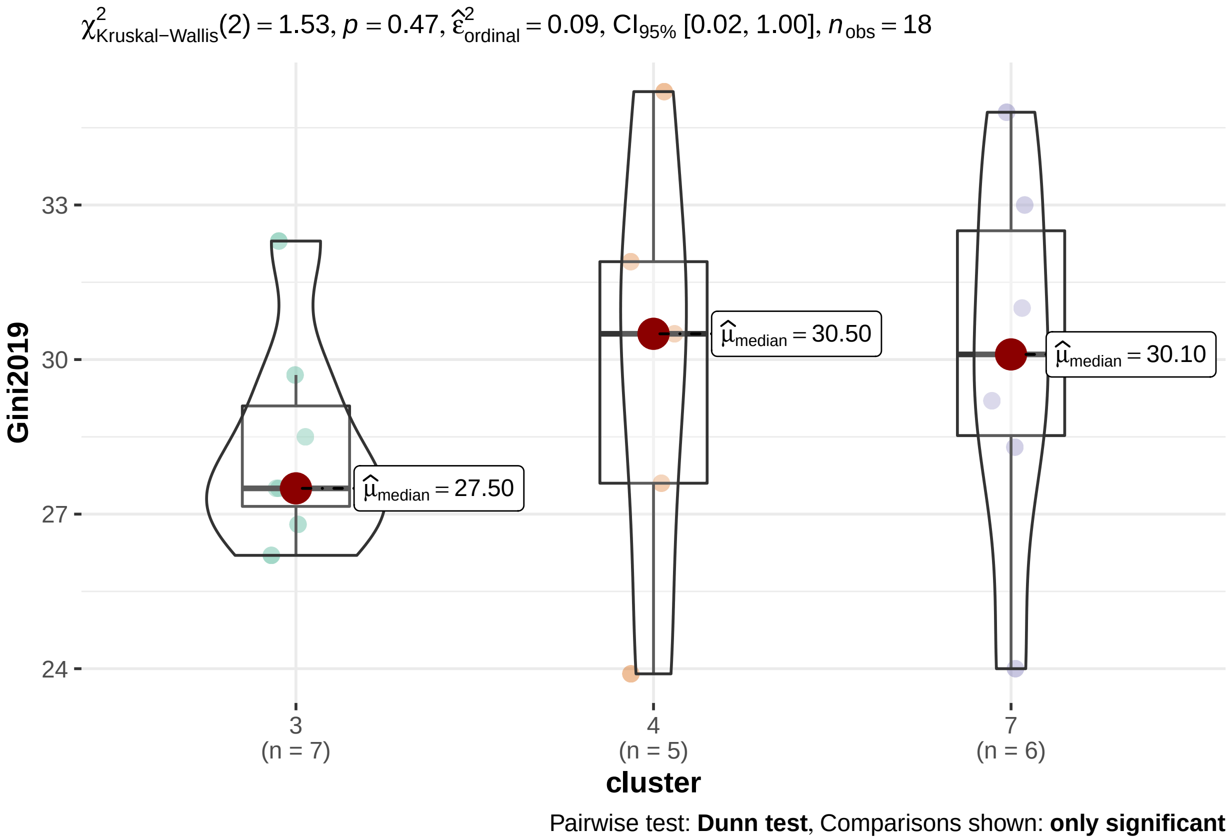

```r
## Gender Pay Gap
ggbetweenstats(
  data = big_clusters,
  x = cluster,
  y = Gender_Pay_Gap,
  type = 'nonparametric',
  var.equal = FALSE,
  pairwise.display = 'all'
)
```



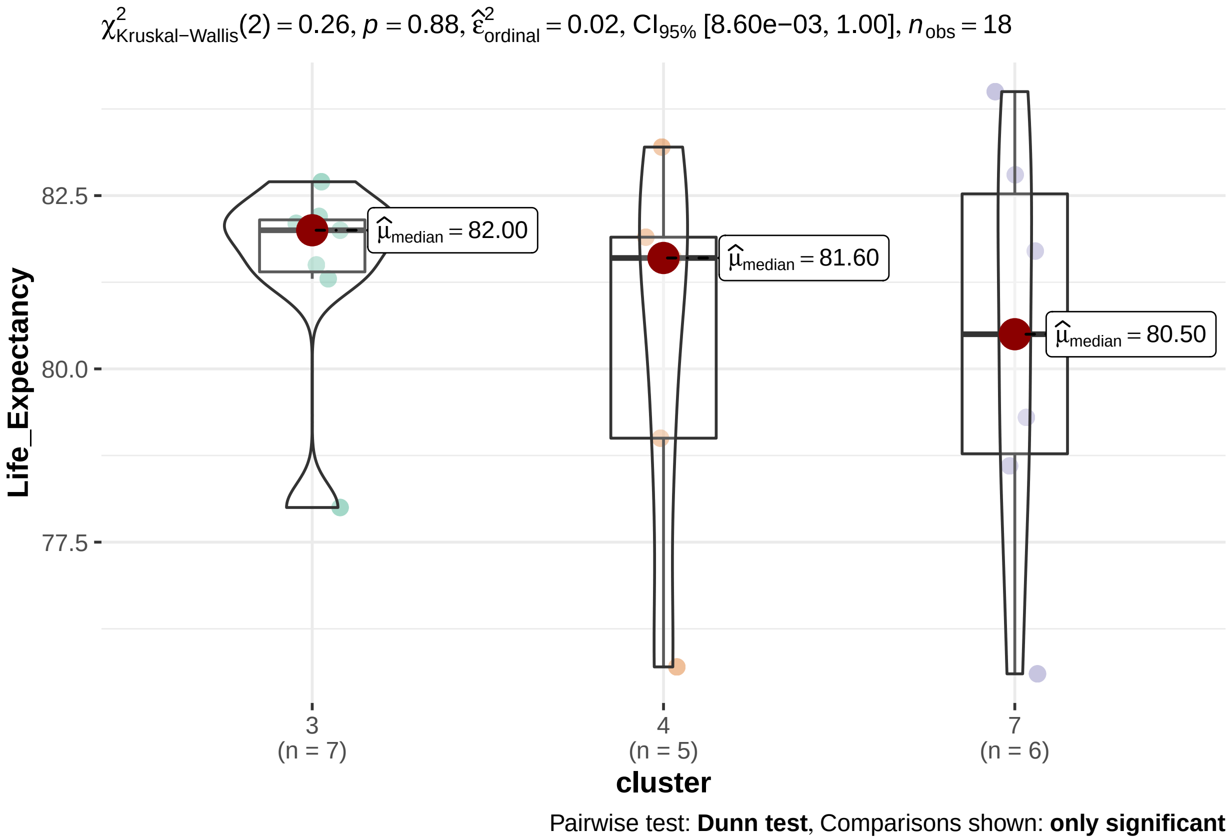

```r
## Population
ggbetweenstats(
  data = big_clusters,
  x = cluster,
  y = Population,
  type = 'nonparametric',
  var.equal = FALSE,
  pairwise.display = 'all'
)
```




```
## Gini2019
ggbetweenstats(
  data = big_clusters,
  x = cluster,
  y = Gini2019,
  type = 'nonparametric',
  var.equal = FALSE
)
```




```
## Life_Expectancy
ggbetweenstats(
  data = big_clusters,
  x = cluster,
  y = Life_Expectancy,
  type = 'nonparametric',
  var.equal = FALSE
)
```



$\chi^2_{\text{Kruskal-Wallis}}(2) = 0.26$, $p = 0.88$, $\widehat{\varepsilon}^2_{\text{ordinal}} = 0.02$, $\text{CI}_{95\%}$ [8.60e−03, 1.00], $n_{\text{obs}} = 18$
Life_Expectancy
82.5
80.0
77.5
$\widehat{\mu}_{\text{median}} = 82.00$
$\widehat{\mu}_{\text{median}} = 81.60$
$\widehat{\mu}_{\text{median}} = 80.50$
3
(n = 7)
4
(n = 5)
7
(n = 6)
cluster
Pairwise test: **Dunn test**, Comparisons shown: **only significant**