Mapping Organizational Culture Schemas Based On Correlational Class Analysis: A Tutorial

Readme: CCA Application Instructions in R Package

Purpose: In this readme file, we detail all the codes needed to run the correlational class analysis (CCA). For this, we divided the file into 8 steps. In the first four steps, we point out which packages and files are needed to reproduce the tutorial. In step 5, we generate the correlational classes from a set of previously defined variables, illustrating each of the classes with graphs (5.1) and heatmaps (5.2). In step 6, we generate clusters with the same variables used in the CCA, with the intention of demonstrating, in step 7, that the classes present different results than other grouping techniques. Finally, in step 8, we illustrate one application of CCA, dividing the sample according to the classes, running regression models for each one, comparing them. Additionally, we included in Appendix A the items of the scale used to generate the CCA and, in Appendix B, the regression models diagnostics.

1. Step one - How to download and install R and RStudio

To install R, go to: http://cloud.r-project.org/, click the download link for R according to your operating system. Click on the "base" link, then click on the "Download R for ..." link. After the download open and run. Also click on the "Rtools" link and download the recommended version.

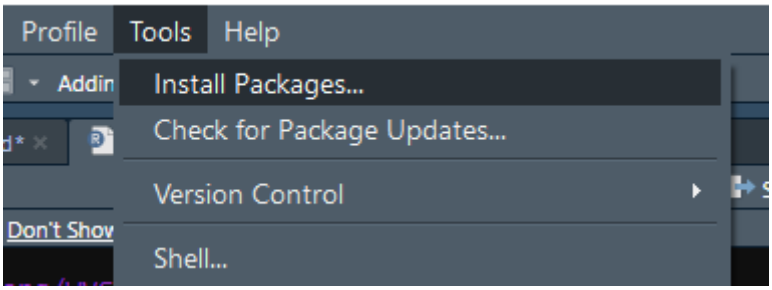
To install RStudio, go to: https://rstudio.com/products/rstudio/download/ and click on the download link for the appropriate version for your interest. To extract files from RStudio in pdf or docx version, it is useful to download and install MikTex as well, visit: https://miktex.org/download, click on the tab that represents your operating system and download it. For more information read the available tutorial.

2. Step two - Installing packages

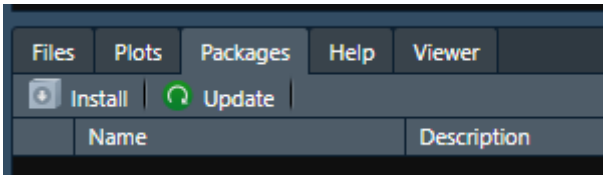
To install packages in RStudio use the *install.packages ()* function and press Ctrl + Enter. The # symbol is used when you want to put comments. Error messages may appear requesting the installation of other packages needed to use the others. Just continue with the installation of it. Package names must be enclosed in quotation marks within parentheses, as in the example of packages used in our analysis below:

```
install.packages("tidyverse") # set of packages "tidy"
install.packages("haven") # package to import data
install.packages("janitor") # package to summarize data
install.packages("readr") # to load data
install.packages("dplyr") # to move organized data
install.packages("ggplot2") # to create and visualize graphics
install.packages("igraph") # to run network graphics and analysis
install.packages("d3heatmap") # to run interactive heatmaps
install.packages("factoextra") # assists in the construction of graphics
install.packages("irr") # to run reliability tests on interrelated variables
install.packages("gmodels") # package with tools for adjusting models
install.packages("psych") # package with quantitative tools
install.packages("jtools") # package for data analysis and presentation
install.packages("stargazer") # package for regression formatting
install.packages("huxtable") # package to assist in formatting tables
install.packages("gap") # we will use to apply the Chow test
```

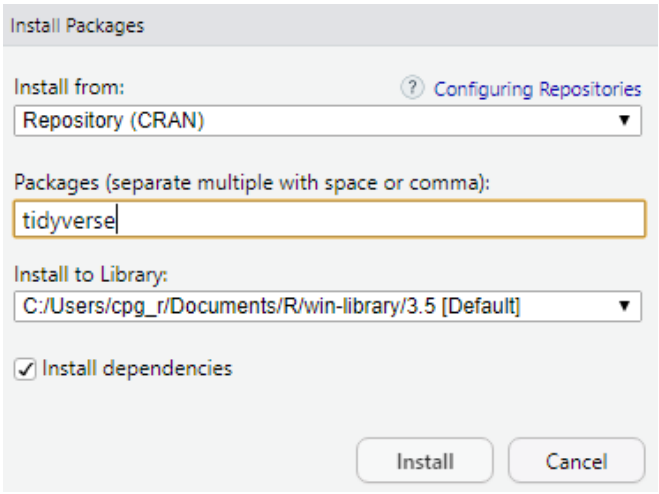
Another way to install packages is to click on the top tab "Tools" and then "Install Packages", as in the following image:



It can also be done by clicking on the "Packages" tab (bottom right corner) and then on the "Install" button.



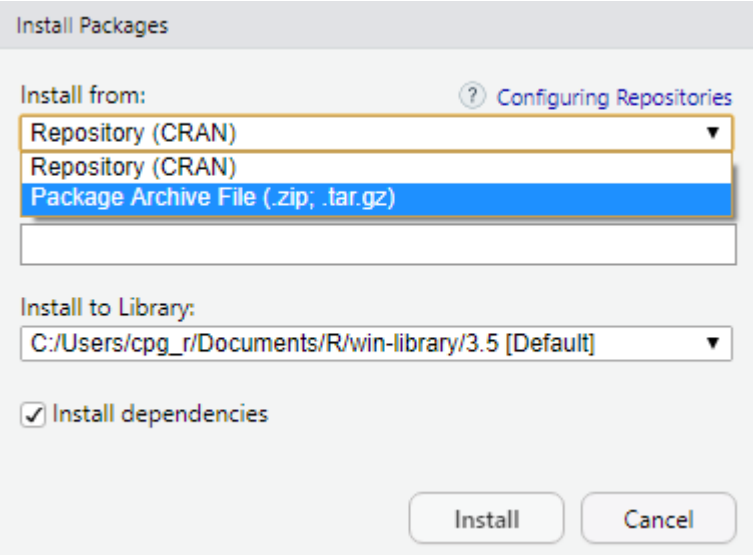
In the tab that will open, just write the name of the desired package in “Packages (separate multiple with space or comma):” and click on “Install”. The option “Install dependencies” must be activated (see the figure below).



2.1. Corclass Package

Some packages, to be installed, will not be available in the Repository (CRAN), they must be downloaded in specific links and then uploaded to RStudio following the previous steps. However, it must be changed from Repository (CRAN) to Package Archive File (.zip;.tar.gz), as in the following image. The *corclass* package, used to generate the correlational class analysis (CCA), must be downloaded from the link https://cran.r-project.org/src/contrib/Archive/corclass/ in which the latest version is suggested, from January 20, 2016: corclass_0.1.1.tar.gz

Select the file in the folder where the corclass 0.1.1.tar.gz file was downloaded and click "Install".



3. Step three - Loading the packages

After installing the packages, to load them, use the *library ( )* function with the package name and press Ctrl + Enter.

```
library(tidyverse) # set of packages "tidy"
library(haven) # package to import data
library(janitor) # package to summarize data
library(readr) # to load data
library(dplyr) # to move organized data
library(corclass) # to partition of the data into separate modules
library(ggplot2) # to create and visualize graphics
library(igraph) # to run network graphics and analysis
library(d3heatmap) # to run interactive heatmaps
library(factoextra) # assists in the construction of graphics
library(irr) # to run reliability tests on interrelated variables
library(gmodels) # package with tools for adjusting models
library(psych) # package with quantitative tools
library(jtools) # package for data analysis and presentation
library(stargazer) # package for regression formatting
library(huxtable) # package to assist in formatting tables
library(gap) # we will use to apply the Chow test
```

4. Step four - Loading the database

Before loading the analysis database, make sure that the work directory is allocated. Use the *getwd ( )* function to have the directory address appear on the console. You can download and allocate the database in this directory, or use the *setwd ( )* function to change the directory. The directory address must be enclosed in quotation marks as in the example below:

```
getwd()
```

```
[1] "C:/Users/Dropbox (Pessoal)/Artigos"
```

```
setwd("C:/Users/Dropbox (Pessoal)/RAC")
```

The database is a SPSS file with a .sav extension, to upload to RStudio we will use the *read_sav( )* function of the *haven* package. The file name must be enclosed in quotation marks in the same way as it was named. The database will only be loaded if the file is in the same directory.

```
Data <- read_sav("Organizational_Culture_CCA.sav")
```

You can explore the database using the *glimpse( )* function in the *dplyr* package.

5. Step five - Generating the correlational class

Selecting the variables to run the correlational class. Forming a new object.

```
Data_Class <- select(Data, c(PCoop1,
                             PCoop2,
                             PCoop3,
                             PCoop4,
                             PCoop6,
                             PCoop7,
                             PCoop8,
                             PCoop9,
                             PCoop10,
                             PCoop11,
                             PCoop12,
                             PCoop13,
                             PCoop14,
                             PCoop15,
                             PMerit16,
                             PCoop17,
                             PMerit18,
                             PCoop19,
                             PCoop20,
                             PMerit22,
                             PMerit23,
                             Rig24,
                             Rig25,
                             Rig26,
                             Rig27,
                             Rig28,
                             Rig29,
                             Rig30,
                             Rig31,
                             Rig32,
                             Rig33,
                             Rig34,
                             PMerit35,
                             Rig36,
                             PMerit40,
                             PMerit41,
                             Rig42,
                             Value45,
                             Value46,
                             Value47,
                             Value48,
                             Value49,
                             Value50,
                             Value52,
                             Value53,
                             Value54,
                             Value55))
```

Running the CCA with 0.05 of significance to create the classes based on the data, a message will appear with the number of schematic classes found and their respective sizes. To see, just use the *print( )* function of the base R.

The first argument of the *cca( )* function is the data frame with the variables that will be used to partition the classes according to their schemas. The *filter.significance* argument makes the links insignificant at 0 to increase the accuracy. To disable, just use *filter.significance* = FALSE. The third argument, *filter.value*, represents the significance cutoff adopted for the correlations. *Zero.action* is related to the partitioning graph. Using "drop", the CCA leaves out the lines with 0 of variance. If you use "ownclass", the correlations with 0 of variance are set to 1, which creates a "zero class". Finally, the argument *verbose* = TRUE prints the CCA details on the screen.

```
cca05 <- cca(Data_Class, filter.significance = TRUE, filter.value =0.05, zero.ac
tion = c("drop"), verbose = TRUE)
```

```
## Filtering out correlations for which Pr(|r| != 0) >  0.05
## CCA found 2 schematic classes. Sizes: 105 102
```

Defining the class of each case.

```
ccacases05 <- cca05$membership
```

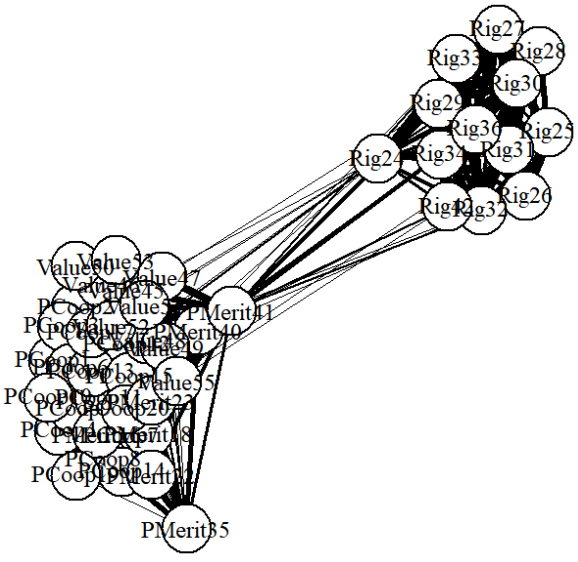
Adding the classes variable (CCA) to the database.

```
Data <- Data %>%
  mutate(CCA = (ccacases05))
```

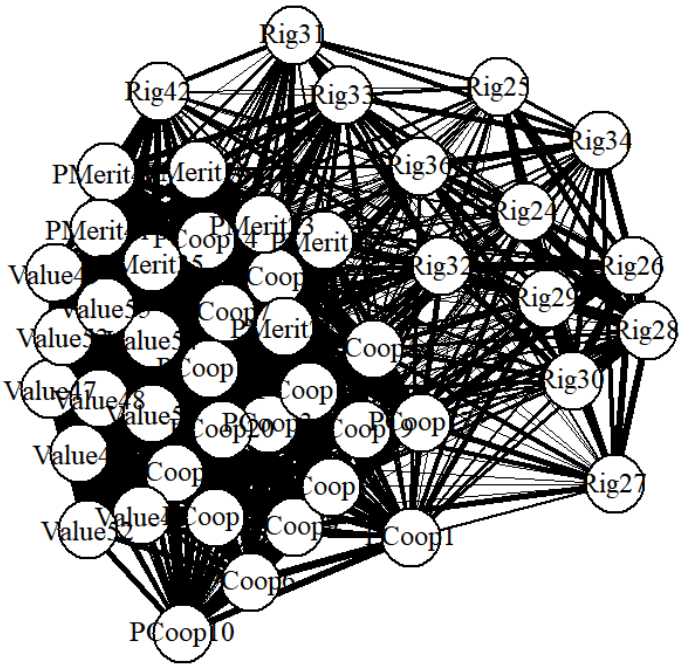
5.1. Plotting the network graphs for each Class

```
res1 <- cca05
plot(res1, 1, cutoff = 0.05, layout.fruchterman.reingold,
drop.neg.ties.for.layout = TRUE, main = 'Class Reaticves', bw = T)
plot(res1, 2, cutoff = 0.05, layout.fruchterman.reingold,
drop.neg.ties.for.layout = TRUE, main = 'Class Resilient', bw = T)
```

Reactive Class



Resilient Class



5.2. Generating interactive heatmaps for each class

To create the Heatmaps, we use the *d3heatmap* package, which generates heat maps for each class according to the correlation of its variables. Hovering the mouse over the map in the html or RStudio version, you can see the correlation value.

First, we will make the correlation matrix of the items used to form the classes.

```
MatrixCor <- cor(Data_Class, use = "all.obs", method = "pearson")
```

In the second moment, we analyze the correlation of each of the classes separately. First, it is necessary to split the base, separating the data from class 1 and class 2.

```
Data_Class <-  Data_Class %>%
  mutate(Class = (ccacases05))
Data_Class1 <- subset(Data_Class, Class=="1", -  c(Class))
Data_Class2 <- subset(Data_Class, Class=="2", -  c(Class))
MatrixCor1 <- cor(Data_Class1, use = "all.obs", method = "pearson")
MatrixCor2 <- cor(Data_Class2, use = "all.obs", method = "pearson")
```

Testing the equality of correlation matrices between classes and between the total sample by Jennrich's test (1970).

```
cortest.jennrich(MatrixCor1, MatrixCor2, n1= 105, n2= 102)
```

```
## $chi2
## [1] 2847.308
##
## $prob
## [1] 6.463781e-159
```

```
cortest.jennrich(MatrixCor1, MatrixCor, n1= 105, n2= 207)
```

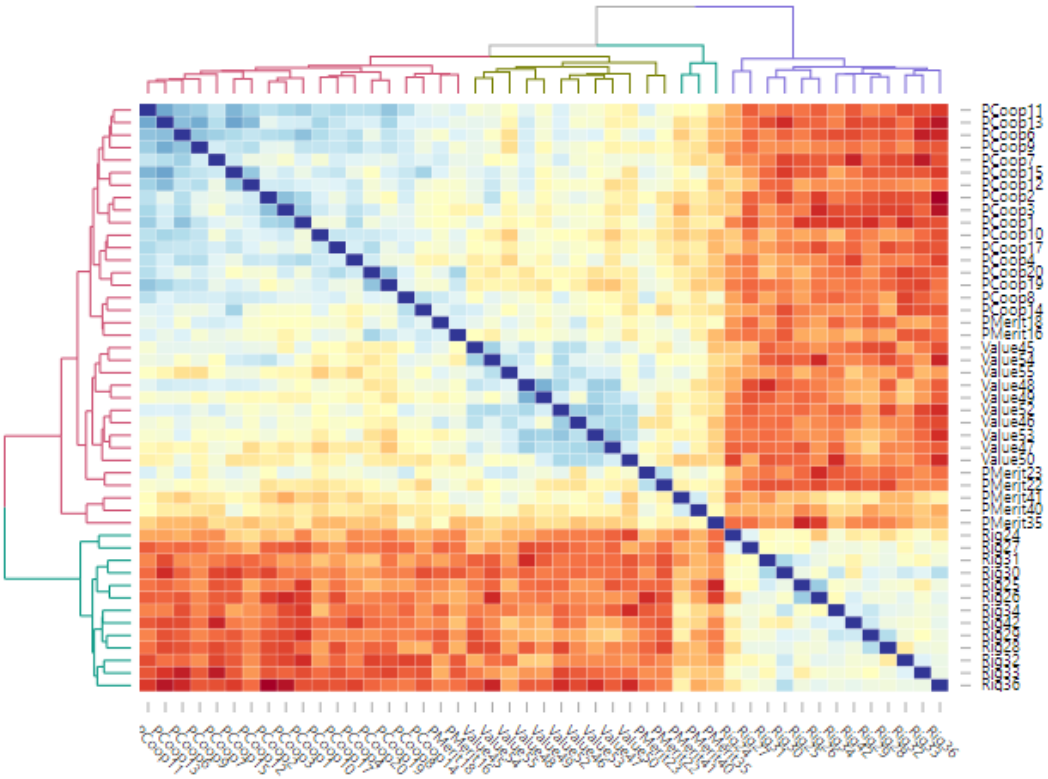
```
## $chi2
## [1] 1379.843
##
## $prob
## [1] 1.522246e-09
```

```
cortest.jennrich(MatrixCor2, MatrixCor, n1= 102, n2= 207)
```

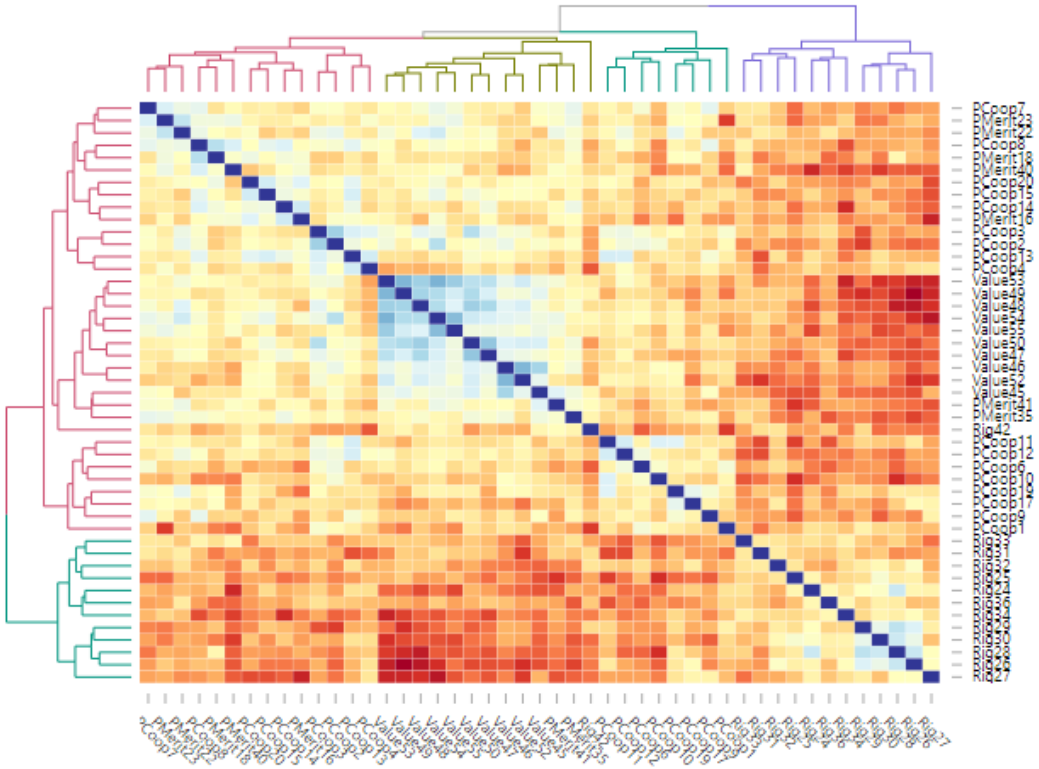
```
## $chi2
## [1] 1315.133
##
## $prob
## [1] 1.145176e-06
```

Interactive heatmap of the class 1 and class 2 data correlation matrix. The strongly negative correlations are red and the strongly positive correlations are blue. The graphics, when run in R Script, appeared in the “Viewer” in the quadrant below the Enviroment.

```
d3heatmap((MatrixCor1), colors = "RdYlBu", k_row = 2, k_col = 4)
```



```
d3heatmap((MatrixCor2), colors = "RdYlBu", k_row = 2, k_col = 4)
```



6. Step six - Generating Cluster

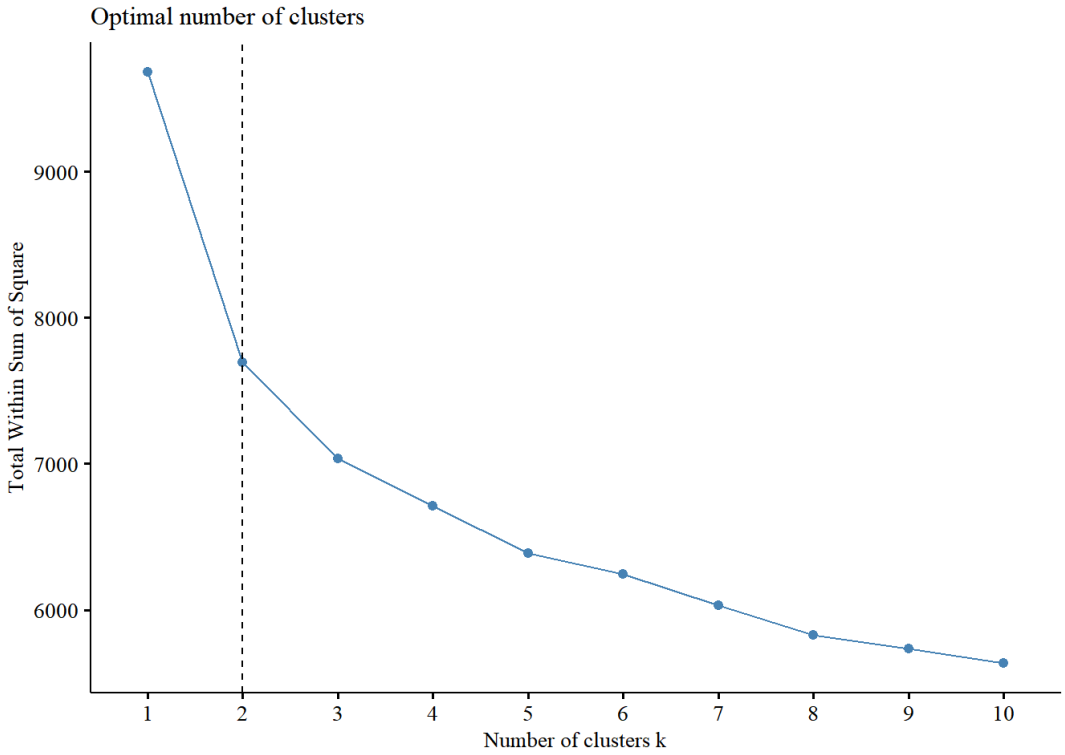
In order to demonstrate the difference between the CCA and cluster technique, we performed a cluster analysis on the same basis and crossed with the chi-square and Kappa tests to point out the difference between both.

Running the cluster technique with Euclidean distance using the Ward method:

```
Data_Class <-  subset(Data_Class, select =  -  c(Class))
dd <- dist(scale(Data_Class), method = "euclidean")
hc <- hclust(dd, method = "ward.D2")
```

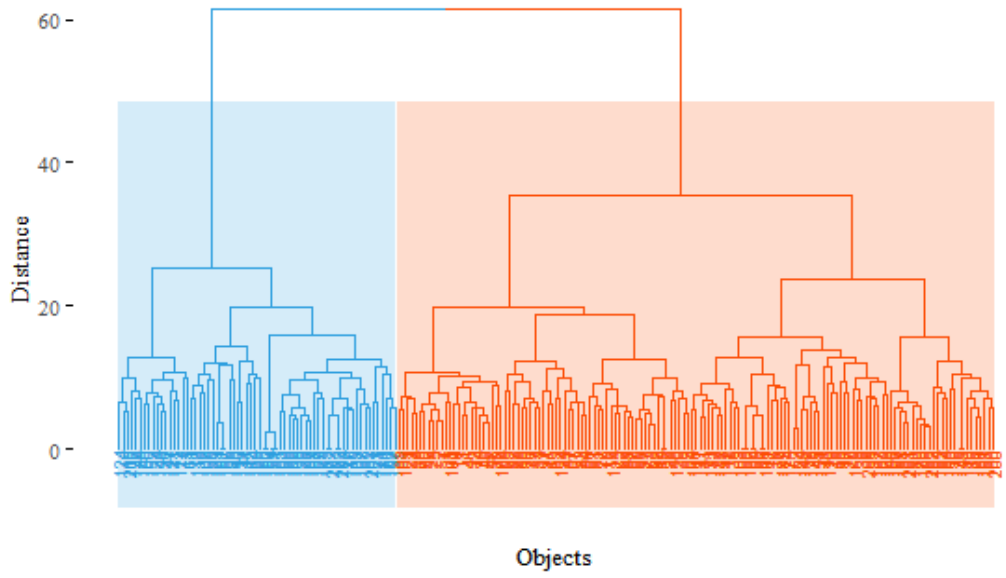
Scree Plot

```
df <- scale(Data_Class)
fviz_nbclust(df, kmeans, method = "wss")  +
geom_vline(xintercept = 2, linetype = 2)
```



Presenting the Dendrogram chart.

```
fviz_dend(hc, k = 2,
cex = 0.5,
main = "Dendrogram - Ward",
          xlab = "Objects", ylab = "Distance", sub = "",
k_colors = c("#2E9FDF", "#FC4E07"),
color_labels_by_k = TRUE,
rect = TRUE,
rect_border = c("#2E9FDF", "#FC4E07"),
rect_fill = TRUE)
```



Performing the grouping of cases in 2 clusters using the K-means method.

```
set.seed(123)
km.res <- kmeans(df, 2, nstart = 25)
```

Defining the cases of each cluster.

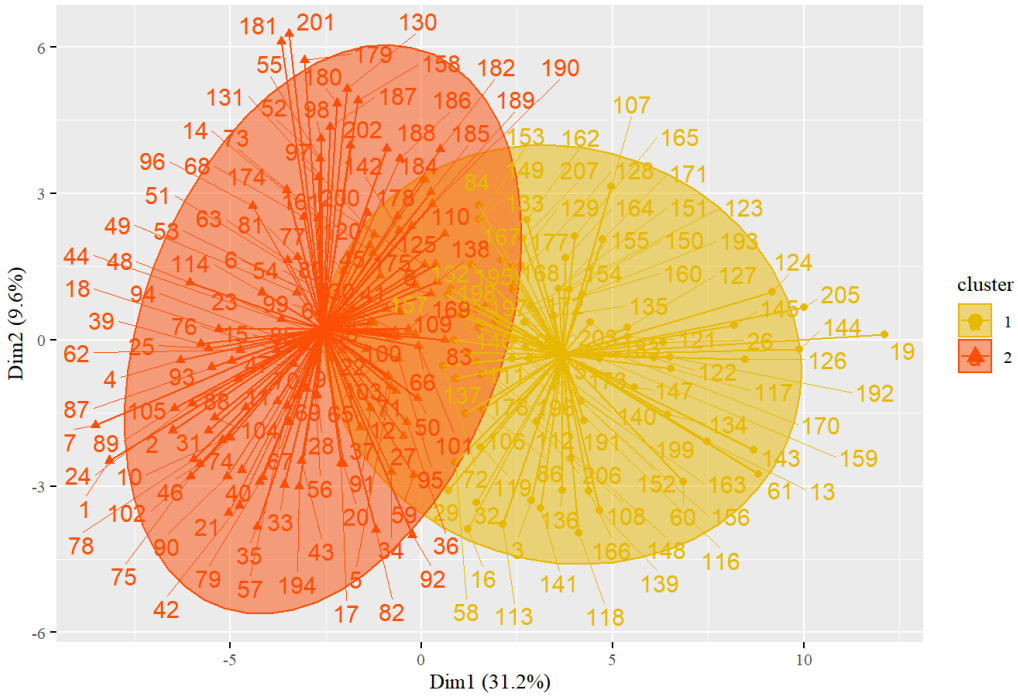
```
cluster <- km.res$cluster
```

Adding the cluster variable to the database.

```
Data <- Data %>%
  mutate(Cluster = (cluster))
```

Cluster plot

```
fviz_cluster(km.res, data = Data_Class,
             axes = c(1, 2),
palette = c("#E7B800", "#FC4E07"),
ellipse.type = "norm",
ellipse.alpha = 0.5,
star.plot = TRUE,
repel = TRUE,
ggtheme = theme_minimal()
)
```



7. Step seven - CCA x Cluster

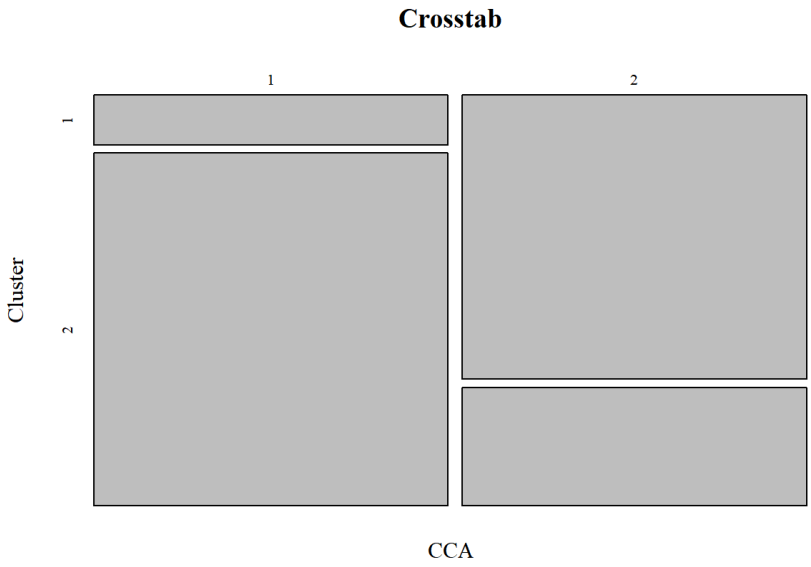
In this step, we seek to demonstrate that the CCA technique generates different results from the cluster technique, as discussed before.

We started with a cross table between the clusters and the classes, followed by the chi-square test, which indicates a differentiation between the categories.

```
Crosstab <- with(Data, table(CCA, Cluster))
round(prop.table(Crosstab, margin = 2), 2)
```

```
##    Cluster
## CCA    1    2
##   1 0.75 0.15
##   2 0.25 0.85
```

```
plot(Crosstab)
```




```
chisq.test(Crosstab)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  Crosstab
## X-squared = 70.048, df = 1, p-value < 2.2e-16
```

Selecting the variables representing the Cluster and CCA to apply the Cohen's Kappa test that presented a moderate result for the variables CCA and Cluster.

```
Kapa <- subset(Data, select = c(CCA, Cluster))
kappa2(Kapa, weight = "equal")
```

```
##  Cohen's Kappa for 2 Raters (Weights: equal)
##
##  Subjects = 207
##    Raters = 2
##     Kappa = 0.583
##
##         z = 8.51
##   p-value = 0
```

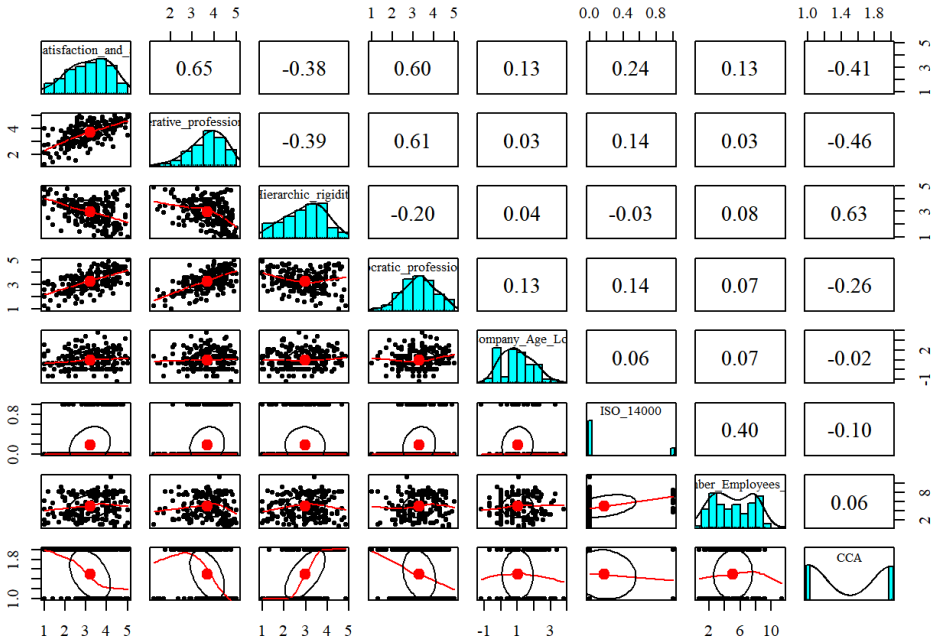
8. Step eighth – Linear regression analysis for each class

As seen, correlational class analysis is a method for grouping cases that share the same schemas into discrete groups. CCA only classifies, does not test hypotheses directly. However, we can use such groups to divide the sample into different classes, comparing the effect of a set of independent variables on a dependent one. For this, we select variables for a linear model (regression) analysis. For this, we ran a pair of linear regressions comparing the results between classes using the Chow test.

```
Data.R <- subset(Data, select= c(Employee_satisfaction_and_appreciation,
                                 Cooperative_professionalism,
                                 Hierarchic_rigidity,
                                 Meritocratic_professionalism,
                                 Company_Age_Log, ISO_14000,
                                 Number_Employees_Log,
                                 CCA)
                                 )
```

Diagnosis of variables, analyzing: normality, linearity, multicollinearity, extreme values (outliers) and homoscedasticity.

```
pairs.panels(Data.R, col="red")
```



Split the data according to the class.

```
Data.R_Class1 <- subset(Data.R, CCA == "1")
Data.R_Class2 <- subset(Data.R, CCA == "2")
```

To make the script easier to read, we select the variables for analysis by removing the extreme cases. To see the step-by-step details, see appendix B.

```
Data.R_Class1 <- Data.R_Class1[-c(5, 15, 16, 20, 29, 48, 52, 55, 58, 59, 61, 77
, 80, 85, 90, 91, 92, 95, 102),]

Data.R_Class2 <- Data.R_Class2[-c(7, 18, 20, 61, 65, 69, 70, 77),]
```

Running multiple regressions for the variables: Employee_satisfaction_and_appreciation, Cooperative_professionalism, Hierarchic_rigidity, Meritocratic_professionalism, Company_Age_Log, Number_Employees_Log, ISO_14000. Class 1 only.

```
fit1 <- lm(Employee_satisfaction_and_appreciation ~
             Cooperative_professionalism +
             Hierarchic_rigidity +
             Meritocratic_professionalism +
             Company_Age_Log +
             Number_Employees_Log +
             ISO_14000,
           data = Data.R_Class1)
summary(fit1)
```

```
##
## Call:
## lm(formula = Employee_satisfaction_and_appreciation ~ Cooperative_professionalism +
##     Hierarchic_rigidity + Meritocratic_professionalism + Company_Age_Log +
##     Number_Employees_Log + ISO_14000, data = Data.R_Class1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65697 -0.25533  0.03982  0.26402  0.58436
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   2.0208934  0.3929246   5.143 2.69e-06 ***
## Cooperative_professionalism   0.2637233  0.0803351   3.283 0.001657 **
## Hierarchic_rigidity          -0.3611673  0.0637196  -5.668 3.57e-07 ***
## Meritocratic_professionalism  0.3405022  0.0729962   4.665 1.59e-05 ***
## Company_Age_Log               0.1410709  0.0479451   2.942 0.004511 **
## Number_Employees_Log          0.0000177  0.0215958   0.001 0.999349
## ISO_14000                     0.5047420  0.1252284   4.031 0.000148 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3622 on 65 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.7691, Adjusted R-squared:  0.7478
## F-statistic: 36.09 on 6 and 65 DF,  p-value: < 2.2e-16
```

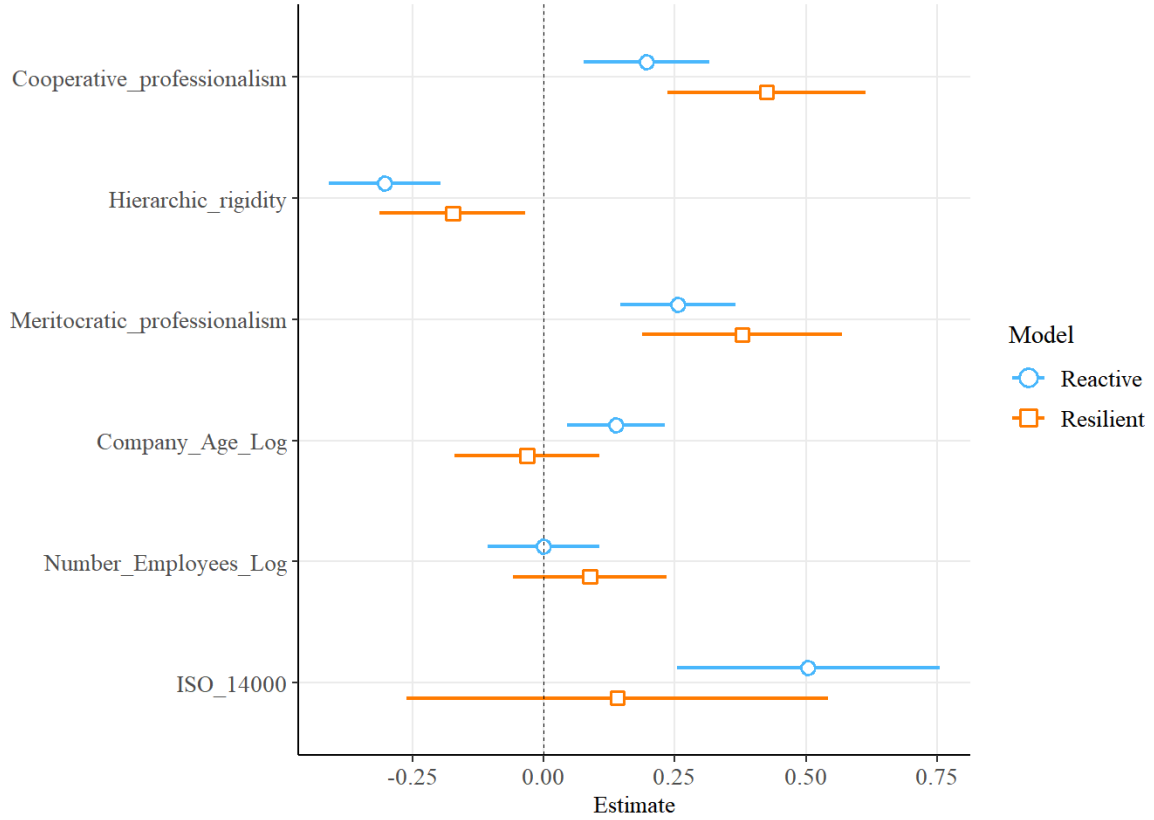
Running multiple regressions for the variables: Employee_satisfaction_and_appreciation, Cooperative_professionalism, Hierarchic_rigidity, Meritocratic_professionalism, Company_Age_Log, Number_Employees_Log, ISO_14000. Class 2 only.

```
fit2 <- lm(Employee_satisfaction_and_appreciation ~
             Cooperative_professionalism +
             Hierarchic_rigidity +
             Meritocratic_professionalism +
             Company_Age_Log +
             Number_Employees_Log +
             ISO_14000,
           data = Data.R_Class2)
summary(fit2)
```

```
##
## Call:
## lm(formula = Employee_satisfaction_and_appreciation ~ Cooperative_professionalism +
##     Hierarchic_rigidity + Meritocratic_professionalism + Company_Age_Log +
##     Number_Employees_Log + ISO_14000, data = Data.R_Class2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26597 -0.35030  0.07777  0.44791  0.95034
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -0.02505    0.47101  -0.053 0.957727
## Cooperative_professionalism   0.62514    0.13890   4.501 2.47e-05 ***
## Hierarchic_rigidity          -0.25893    0.10421  -2.485 0.015229 *
## Meritocratic_professionalism  0.48745    0.12329   3.954 0.000175 ***
## Company_Age_Log              -0.03925    0.08546  -0.459 0.647357
## Number_Employees_Log          0.03666    0.03056   1.200 0.234104
## ISO_14000                     0.14096    0.20165   0.699 0.486723
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6053 on 74 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.6262, Adjusted R-squared:  0.5959
## F-statistic: 20.66 on 6 and 74 DF,  p-value: 4.612e-14
```

At the end you can generate a graph with the coefficients of the models found for class 1 and class 2, using the *plot_summ( )* function of the *jtools* package.

```
plot_summs(fit1, fit2, scale = TRUE, model.names = c("Reactive", "Resilient"))
```



Generating and saving in pdf file, the table with the regression templates found for each class, using the function *export_summs( )*. You can also use the *stargazer* function of the package with the same name. To be able to save the file in pdf or doc, you must have the *MikTex* package installed. If you don't, you can view the table in the console by removing the arguments: "to.file" and "file.name".

```
export_summs(fit1, fit2, error_format = "[{conf.low}, {conf.high}]", to.file =
"pdf", file.name = "Results.pdf")

stargazer(fit1, fit2, type = "html", out = "Results.doc")
```

	Dependent variable:	
	Employee_satisfaction_and_appreciation	
	(1)	(2)
Constant	2.021***	-0.025
	(0.393)	(0.471)
Cooperative_professionalism	0.264***	0.625***
	(0.080)	(0.139)
Hierarchic_rigidity	-0.361***	-0.259**
	(0.064)	(0.104)
Meritocratic_professionalism	0.341***	0.487***
	(0.073)	(0.123)
Company_Age_Log	0.141***	-0.039
	(0.048)	(0.085)
Number_Employees_Log	0.00002	0.037

	(0.022)	(0.031)
ISO_14000	0.505***	0.141
	(0.125)	(0.202)
Observations	72	81
R^2	0.769	0.626
Adjusted R^2	0.748	0.596
Residual Std. Error	0.362 (df = 65)	0.605 (df = 74)
F Statistic	36.087*** (df = 6; 65)	20.662*** (df = 6; 74)
Note:		*p**p***p<0.01

Finally, we apply the Chow test, analyzing whether the two regressions (*Reactive Class* and *Resilient Class*) are structurally different. The result indicates that they are significantly different.

```
Data.R_Class1 <- drop_na(Data.R_Class1)
Data.R_Class2 <- drop_na(Data.R_Class2)
y1 <- subset(Data.R_Class1, select = c(Employee_satisfaction_and_appreciation))
x1 <- subset(Data.R_Class1, select = c(Cooperative_professionalism,
                                       Hierarchic_rigidity,
                                       Meritocratic_professionalism,
                                       Company_Age_Log, Number_Employees_Log,
                                       ISO_14000)
                                      )
y2 <- subset(Data.R_Class2, select=c(Employee_satisfaction_and_appreciation))
x2 <- subset(Data.R_Class2, select = c(Cooperative_professionalism,
                                       Hierarchic_rigidity,
                                       Meritocratic_professionalism,
                                       Company_Age_Log,
                                       Number_Employees_Log,
                                       ISO_14000)
                                      )
chow.test.r <- chow.test(y1,x1,y2,x2)
print(chow.test.r)
```

```
##     F value       d.f.1       d.f.2     P value
##   2.3219394   7.0000000 139.0000000   0.0285767
```

References

Boutyline, A. (2016). *corclass: correlational class analysis.* R package version 0.1. 1.

Csardi, G., & Nepusz, T. (2019). *igraph: Network analysis and visualization.* R package version, 1.2.4.2.

Cheng, J., Galili, T., Bostock, M., & Palmer, J. (2018). *d3heatmap: Interactive Heat Maps Using 'htmlwidgets' and 'D3. Js.'.* R package version, 0.6.1.2.

David Hugh-Jones (2020). *huxtable: Easily Create and Style Tables for LaTeX, HTML and Other Formats*. R package version 4.7.1.

Firke, S. (2020). *janitor: Simple tools for examining and cleaning dirty data.* R package version, 1.2.1.

Garmer, M., Lemon, J., Fellows, I., & Singh, S. (2019). *irr: Various coefficients of interrater reliability and agreement.* R package version, 0.84.1.

Hlavac, M. (2018). *stargazer: Well-formatted regression and summary statistics tables.* R package version, 5.2.2.

Jennrich, R. I. (1970). An asymptotic $\chi 2$ test for the equality of two correlation matrices. *Journal of the American Statistical Association*, 65(330), 904-912.

Kassambara, A. (2019). *factoextra: Extract and visualize the results of multivariate data analysis.* R package version, 1.0.6.

Long, J. A. (2020). *jtools: analysis and presentation of social scientific data.* R package version 2.0.2.

Revelle, W. (2020). *psych: Procedures for psychological, psychometric, and personality research.* R package version, 1.9.12.31

Warnes, G. R., Bolker, B., Lumley, T., & Johnson, R. C. (2018). *gmodels: Various R programming tools for model fitting.* R package version, 2.18.1.

Wickham, H. (2019). *tidyverse: Easily install and load the 'tidyverse'.* R package version, 1.3.0

Wickham, H., & Miller, E. (2019). *haven: Import and export 'SPSS', 'stata' and 'SAS' files.* R package version, 2.2.0.

Wickham, H., Chang, W., Henry, L., Pedersen, T. L., Takahashi, K., Wilke, C., & Woo, K. (2020). *ggplot2: Create elegant data visualisations using the grammar of graphics.* R package version, 3.3.0.

Wickham, H., Francois, R., Henry, L., & Müller, K. (2020). *dplyr: A grammar of data manipulation.* R package version 0.8.5, 3.

Wickham, H., Hester, J., & Francois, R. (2017). *readr: Read rectangular text data*. R package version, 1.3.1.

Zhao, J. H. (2020). *gap: Genetic Analysis Package.* R package version, 1.2.2.

Appendix A – Organizational Values Scale

Items extracted from Ferreira *et al*. (2002)	Factor Loading
Cooperative Professionalism	
PCoop1 Cooperation is more valued than competition.	0,594
PCoop2 Employees' creative ideas are used to achieve better results with fewer resources.	0,569
PCoop3 Individual employee initiatives are encouraged.	0,633
PCoop4 Employee initiatives are valued when they are not against the interests of the companies.	0,649
PCoop6 The spirit of collaboration is considered very important.	0,659
PCoop7 Involvement with the company's ideals is a highly valued attitude.	0,551
PCoop8 Employees who manage to make a career are the ones who make an effort and show willingness to learn.	0,579
PCoop9 The employee's concern with the quality of his service is well regarded.	0,647
PCoop10 Being cordial to colleagues is one of the most encouraged attitudes.	0,577
PCoop11 Effort and dedication to work are qualities that are greatly appreciated.	0,742
PCoop12 The ability to perform a variety of tasks is highly valued.	0,618
PCoop13 The professionalism of employees is seen as a great virtue.	0,729
PCoop14 Professional advancement is a natural consequence of the merit and competence of employees.	0,449
PCoop15 The concern with overcoming day-to-day difficulties is seen as of great value.	0,581
PCoop17 Employees who demonstrate dedication and a spirit of collaboration are the best models to follow.	0,631
PCoop19 The quality of the service performed is considered one of the employee's greatest virtues.	0,677
PCoop20 The employees who "wear the shirt" are the most valued figures within the organization.	0,594
Hierarchical Rigidity and Competition	
Rig24 Professional growth is not usually rewarded financially.	0,535
Rig25 There are no important figures that can serve as examples for employees.	0,662
Rig26 Sends who can, obeys who has judgment.	0,687
Rig27 The difficulty of career advancement leads the company to lose good employees to competitors.	0,641
Rig28 Creativity is not rewarded as it should be.	0,688
Rig29 There is no room for individual employee initiatives.	0,719
Rig30 Opportunities for functional advancement are limited by the company's rigid structure.	0,735
Rig31 Lack of financial resources hinders the valuation of employee welfare.	0,592
Rig32 People prefer to be ordered not to take responsibility.	0,646
Rig33 Excessive concern for welfare is seen as detrimental to the company.	0,586
Rig34 It is very difficult to make a career within the organization.	0,624
Rig36 Professional advancement depends on luck	0,687
Rig42 Competition is valued, even if not in a healthy way, because the company's main objective is productivity and profit.	0,519
Valuing and Satisfying Employees	
Value45 Employee welfare is seen as a way to ensure greater production.	0,561
Value46 The personal needs and well-being of employees are a constant concern of the company.	0,662
Value47 Invests itself in the professional growth of employees.	0,732
Value48 Employees receive training to develop their creativity.	0,701
Value49 Programs to increase employee satisfaction are regularly developed.	0,710
Value50 The aim is to maintain an atmosphere of security and stability to leave employees satisfied and confident.	0,689
Value52 Invests itself in a good working environment in order to ensure the well-being of employees.	0,689
Value53 Programs aimed at improving employee well-being are implemented and tested.	0,765
Value54 Employees' creative ideas are put into practice as a way of making them motivated.	0,574
Value55 Personal goals, when of great value, are incorporated into organizational goals.	0,546
Meritocratic Professionalism	
PMerit16 Employees who make a career quickly are those who "wear the company shirt".	0,433
PMerit18 People who make a career quickly are those who demonstrate the greatest knowledge within their fields.	0,468
PMerit22 Directors who innovate and promote significant change are the real models to be followed.	0,417

PMerit23 Employees who are committed to the mission and ideals of the company become models for other members of the organization.	0,508
PMerit35 Company owners are admired by employees.	0,472
PMerit40 Creativity is one of the basic requirements for the occupation of managerial positions.	0,608
PMerit41 Professional growth is considered indispensable to the employee's permanence in the house.	0,586

Appendix B –Regression Models Diagnostics

Split the data according to the class. To make the script easier to read, we select the variables for analysis by removing the extreme cases.

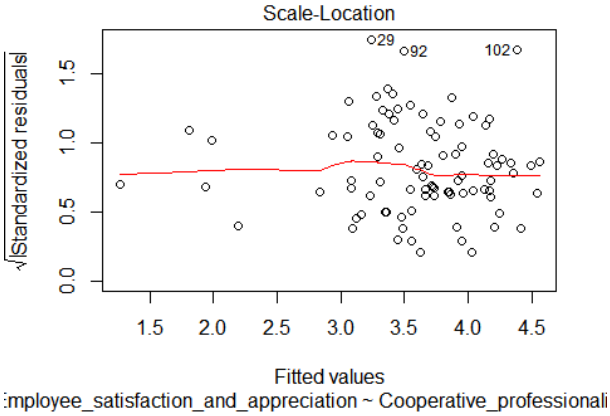
```
Data.R_Class1 <- subset(Data.R, CCA == "1")
Data.R_Class2 <- subset(Data.R, CCA == "2")
```

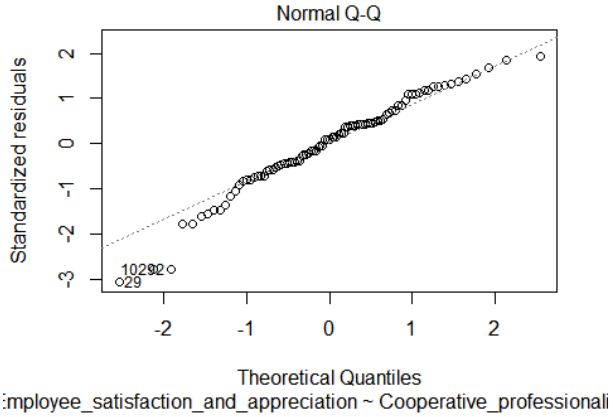
Running multiple regressions for the variables: Employee_satisfaction_and_appreciation, Cooperative_professionalism, Hierarchic_rigidity, Meritocratic_professionalism, Company_Age_Log, Number_Employees_Log, ISO_14000. Class 1 only.

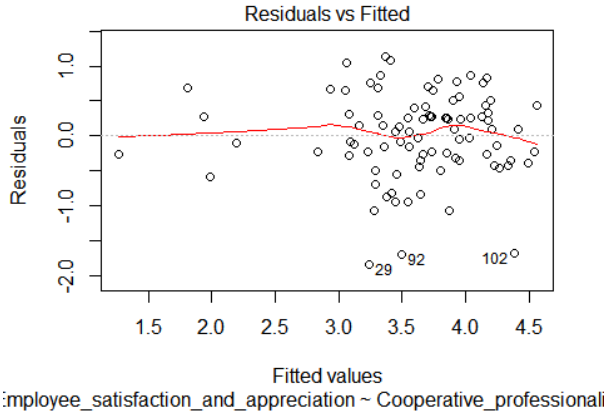
```
fit1 <- lm(Employee_satisfaction_and_appreciation ~
              Cooperative_professionalism +
              Hierarchic_rigidity +
              Meritocratic_professionalism +
              Company_Age_Log +
              Number_Employees_Log +
              ISO_14000,
            data = Data.R_Class1)
summary(fit1)
```

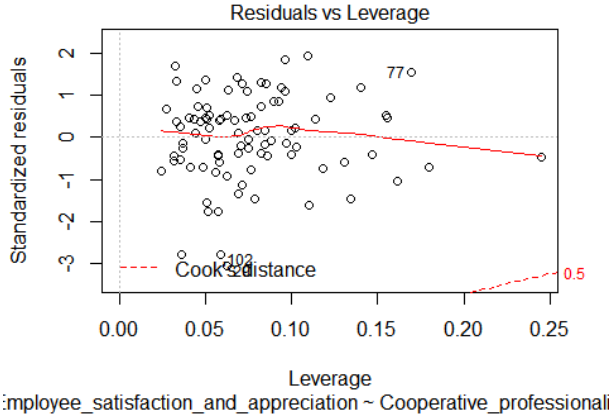
```
##
## Call:
## lm(formula = Employee_satisfaction_and_appreciation ~ Cooperative_professionalism +
##     Hierarchic_rigidity + Meritocratic_professionalism + Company_Age_Log +
##     Number_Employees_Log + ISO_14000, data = Data.R_Class1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8409 -0.3348  0.0549  0.3619  1.1334
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.65265    0.55792   1.170 0.245388
## Cooperative_professionalism  0.40224    0.11085   3.629 0.000488 ***
## Hierarchic_rigidity         -0.09927    0.08946  -1.110 0.270309
## Meritocratic_professionalism 0.37679    0.10170   3.705 0.000378 ***
## Company_Age_Log              0.06429    0.06731   0.955 0.342229
## Number_Employees_Log         0.01894    0.02894   0.655 0.514519
## ISO_14000                    0.33983    0.18375   1.849 0.067910 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6245 on 84 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.4885, Adjusted R-squared:  0.452
## F-statistic: 13.37 on 6 and 84 DF,  p-value: 1.401e-10
```

```
plot(fit1)
```









After some rounds excluding outliers cases, we found the final model. To make it easier, we point out all the cases removed until the final model arrives. This way, we remove the cases: 5, 15, 16, 20, 29, 48, 52, 55, 58, 59, 61, 77, 80, 85, 90, 91, 92, 95, 102.

```
Data.R_Class1 <- Data.R_Class1[-c(5, 15, 16, 20, 29, 48, 52, 55, 58, 59, 61, 77
, 80, 85, 90, 91, 92, 95, 102),]
```

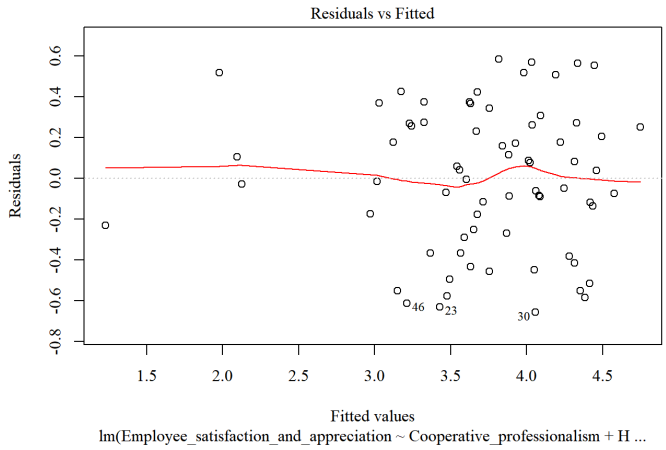
Rodando novamente a regressão retirando os valores extremos.

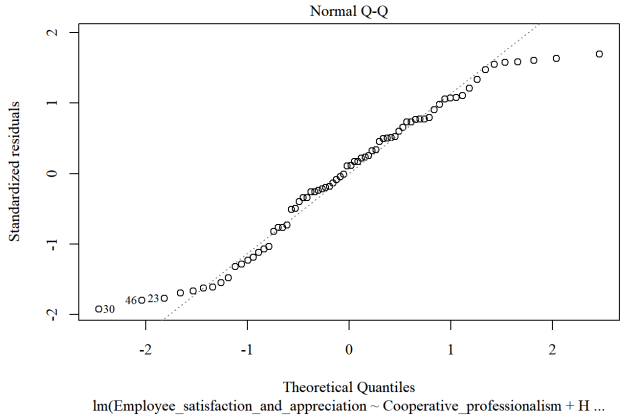
```
fit1 <- lm(Employee_satisfaction_and_appreciation ~
             Cooperative_professionalism +
             Hierarchic_rigidity +
```

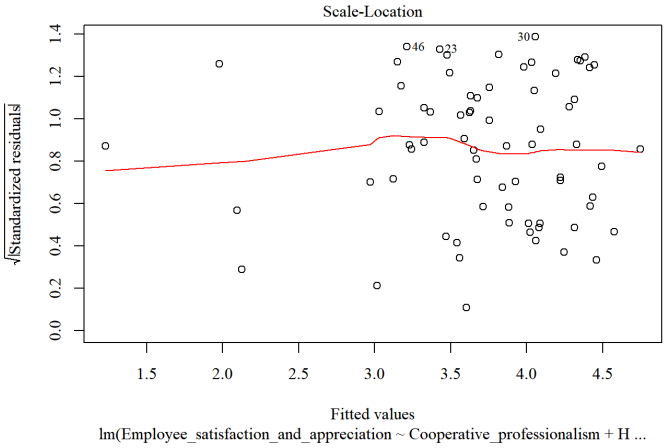
```
              Meritocratic_professionalism +
              Company_Age_Log +
              Number_Employees_Log +
              ISO_14000,
            data = Data.R_Class1)
summary(fit1)
```

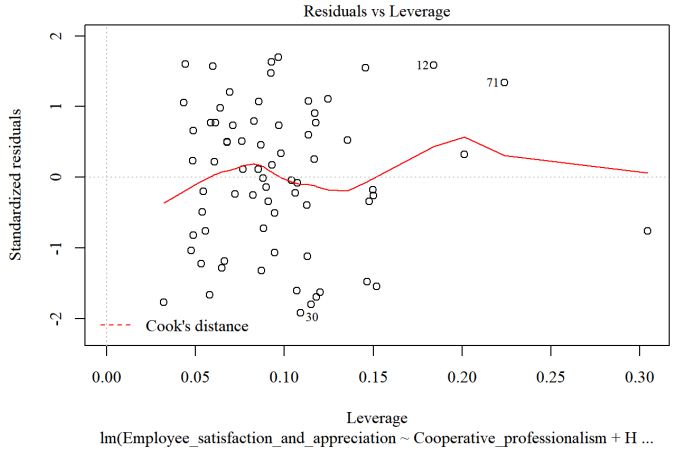
```
##
## Call:
## lm(formula = Employee_satisfaction_and_appreciation ~ Cooperative_professionalism +
##     Hierarchic_rigidity + Meritocratic_professionalism + Company_Age_Log +
##     Number_Employees_Log + ISO_14000, data = Data.R_Class1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.65697 -0.25533  0.03982  0.26402  0.58436
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.0208934  0.3929246   5.143 2.69e-06 ***
## Cooperative_professionalism  0.2637233  0.0803351   3.283 0.001657 **
## Hierarchic_rigidity         -0.3611673  0.0637196  -5.668 3.57e-07 ***
## Meritocratic_professionalism  0.3405022  0.0729962   4.665 1.59e-05 ***
## Company_Age_Log              0.1410709  0.0479451   2.942 0.004511 **
## Number_Employees_Log         0.0000177  0.0215958   0.001 0.999349
## ISO_14000                    0.5047420  0.1252284   4.031 0.000148 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3622 on 65 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.7691, Adjusted R-squared:  0.7478
## F-statistic: 36.09 on 6 and 65 DF,  p-value: < 2.2e-16
```

```
plot(fit1)
```







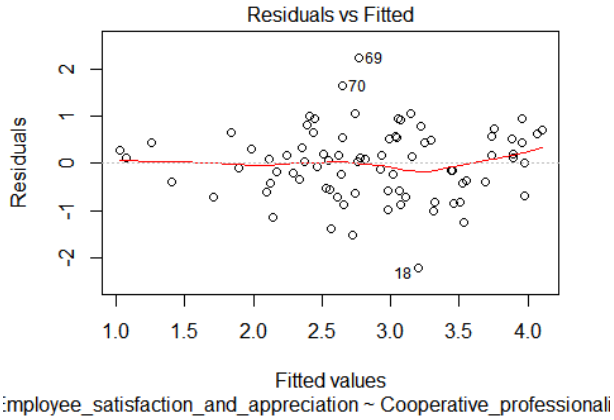


Running multiple regressions for the variables: Employee_satisfaction_and_appreciation, Cooperative_professionalism, Hierarchic_rigidity, Meritocratic_professionalism, Company_Age_Log, Number_Employees_Log, ISO_14000. Class 2 only.

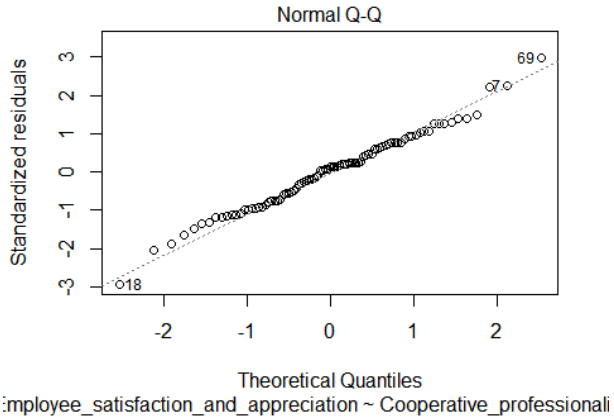
```
fit2 <- lm(Employee_satisfaction_and_appreciation ~
             Cooperative_professionalism +
             Hierarchic_rigidity +
             Meritocratic_professionalism +
             Company_Age_Log +
             Number_Employees_Log +
             ISO_14000,
           data = Data.R_Class2)
summary(fit2)
```

```
##
## Call:
## lm(formula = Employee_satisfaction_and_appreciation ~ Cooperative_professionalism +
##     Hierarchic_rigidity + Meritocratic_professionalism + Company_Age_Log +
##     Number_Employees_Log + ISO_14000, data = Data.R_Class2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.19964 -0.56495  0.08902  0.51493  2.22899
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.02122    0.57456   0.037 0.970627
## Cooperative_professionalism   0.61426    0.16513   3.720 0.000364 ***
## Hierarchic_rigidity          -0.19798    0.12857  -1.540 0.127427
## Meritocratic_professionalism  0.40056    0.14528   2.757 0.007187 **
## Company_Age_Log               0.09410    0.09804   0.960 0.339978
## Number_Employees_Log          0.02624    0.03687   0.711 0.478803
## ISO_14000                     0.13521    0.25204   0.536 0.593087
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7686 on 82 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.4577, Adjusted R-squared:  0.4181
## F-statistic: 11.54 on 6 and 82 DF,  p-value: 2.536e-09
```

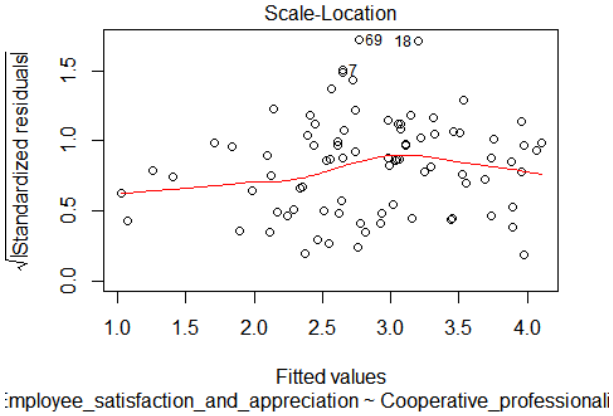
```
plot(fit2)
```

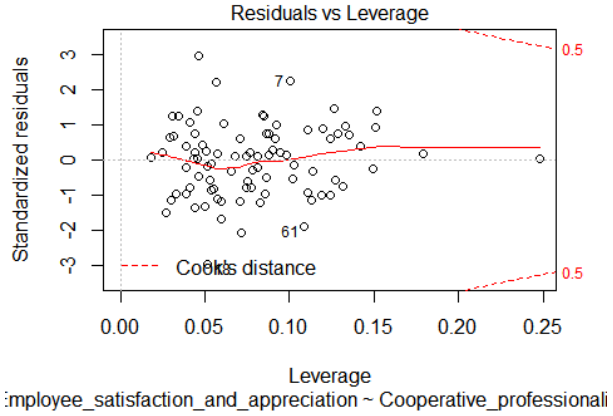


:mployee_satisfaction_and_appreciation ~ Cooperative_professionali



:mployee_satisfaction_and_appreciation ~ Cooperative_professionali





After two rounds excluding outliers from each round, we find the final model. To make it easier to point out all the cases removed until the final model arrives. This way, we remove the cases: 7, 18, 20, 61, 66, 69, 70, 77.

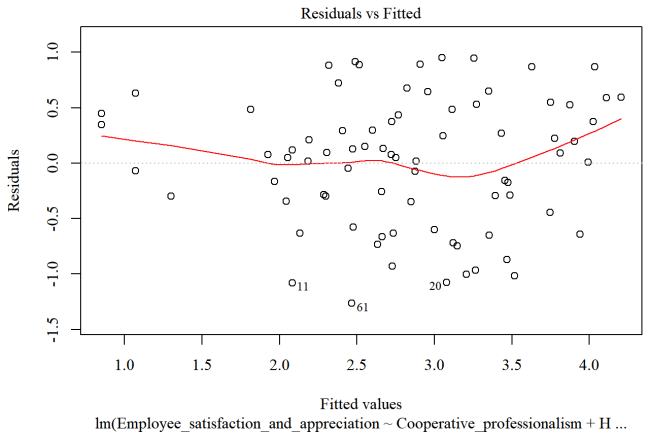
```
Data.R_Class2 <- Data.R_Class2[-c(7, 18, 20, 61, 65, 69, 70, 77),]
```

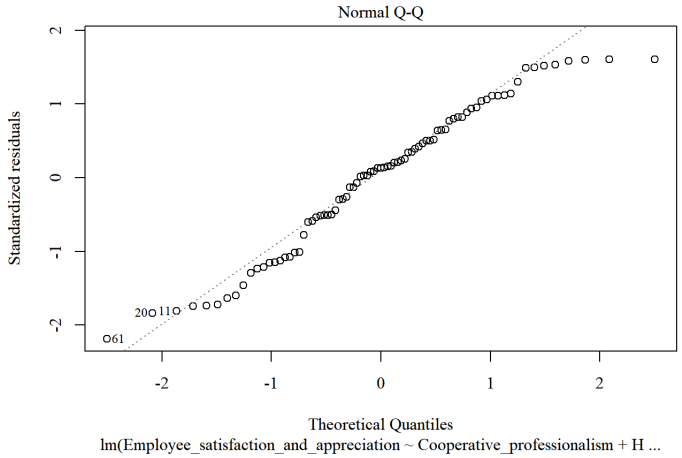
Rotating the regression again by removing the extreme values.

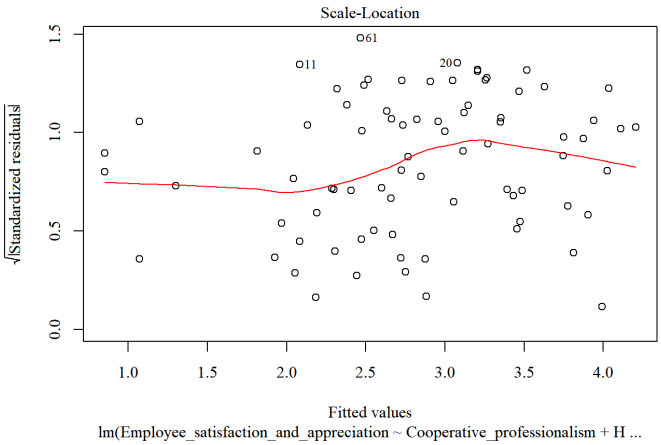
```
fit2 <- lm(Employee_satisfaction_and_appreciation ~
             Cooperative_professionalism +
             Hierarchic_rigidity +
             Meritocratic_professionalism +
             Company_Age_Log +
             Number_Employees_Log +
             ISO_14000,
           data = Data.R_Class2)
summary(fit2)
```

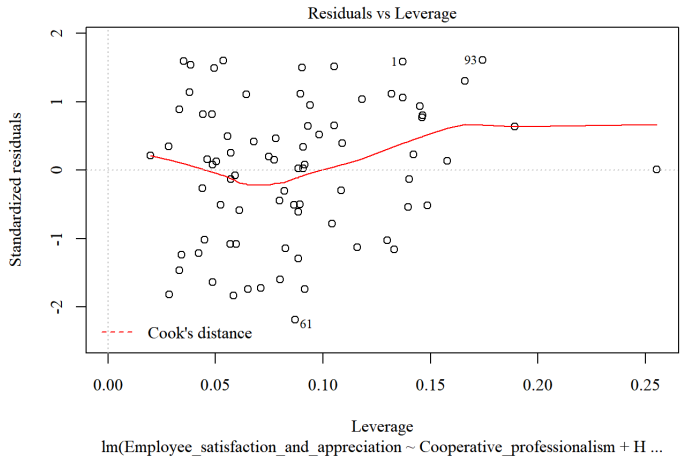
```
##
## Call:
## lm(formula = Employee_satisfaction_and_appreciation ~ Cooperative_professionalism +
##     Hierarchic_rigidity + Meritocratic_professionalism + Company_Age_Log +
##     Number_Employees_Log + ISO_14000, data = Data.R_Class2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26597 -0.35030  0.07777  0.44791  0.95034
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -0.02505    0.47101  -0.053 0.957727
## Cooperative_professionalism  0.62514    0.13890   4.501 2.47e-05 ***
## Hierarchic_rigidity         -0.25893    0.10421  -2.485 0.015229 *
## Meritocratic_professionalism 0.48745    0.12329   3.954 0.000175 ***
## Company_Age_Log             -0.03925    0.08546  -0.459 0.647357
## Number_Employees_Log         0.03666    0.03056   1.200 0.234104
## ISO_14000                    0.14096    0.20165   0.699 0.486723
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6053 on 74 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.6262, Adjusted R-squared:  0.5959
## F-statistic: 20.66 on 6 and 74 DF,  p-value: 4.612e-14
```

```r
plot(fit2)
```






Scale-Location
√|Standardized residuals|
Fitted values
lm(Employee_satisfaction_and_appreciation ~ Cooperative_professionalism + H ...


Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
lm(Employee_satisfaction_and_appreciation ~ Cooperative_professionalism + H ...