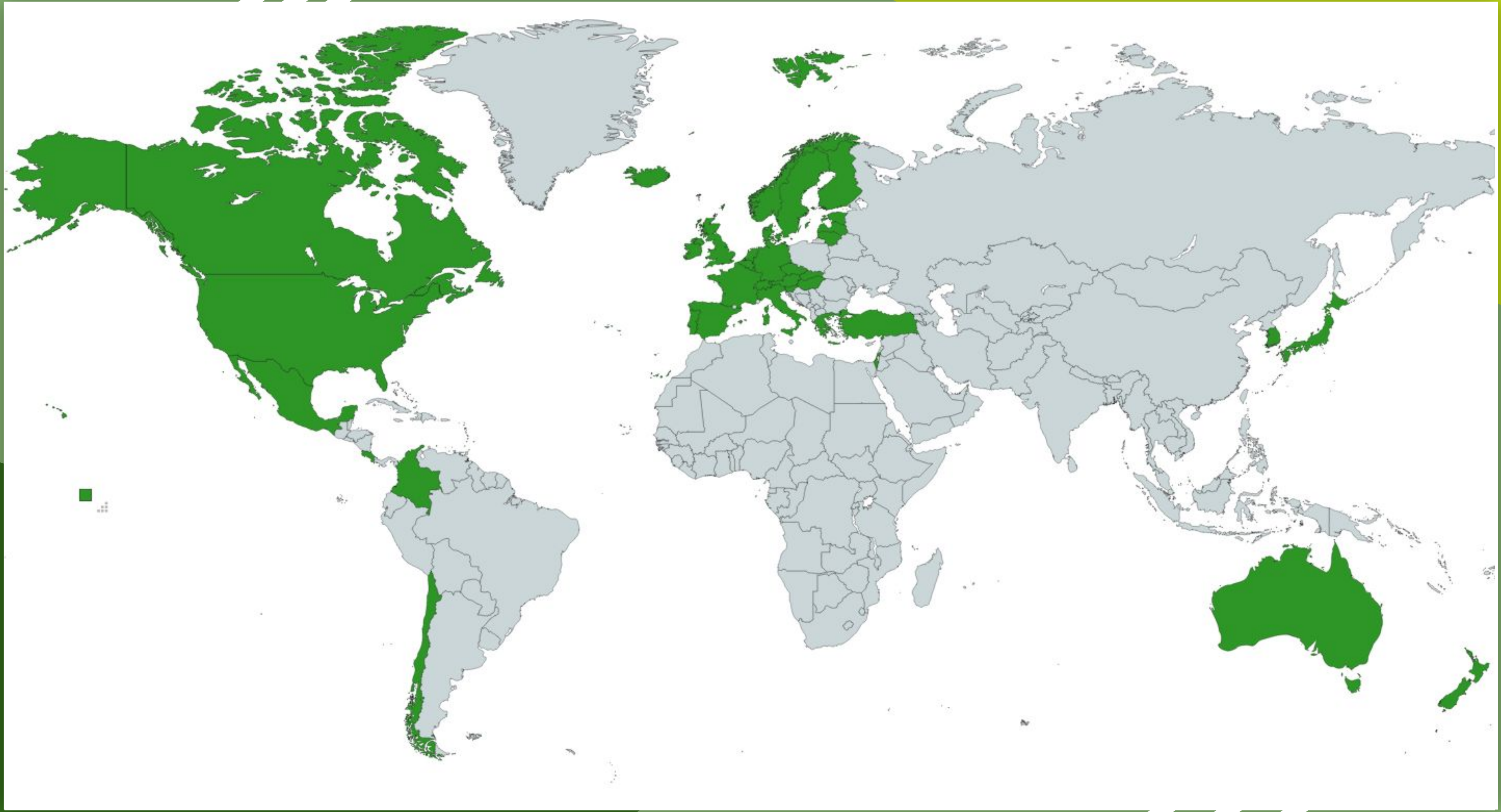
GREEN INVESTMENT CLASSIFICATOR

Giuseppe Grosso
Matteo Fercia



Which country should we choose?

Social media

Economy

Energy

Pollution

Social Media

- Scraped tweets:
  - ➡ Positive
  - ➡ Negative
  - ➡ Neutral

Energy

- Total production
- RE production
- % RE
- Avg consumption

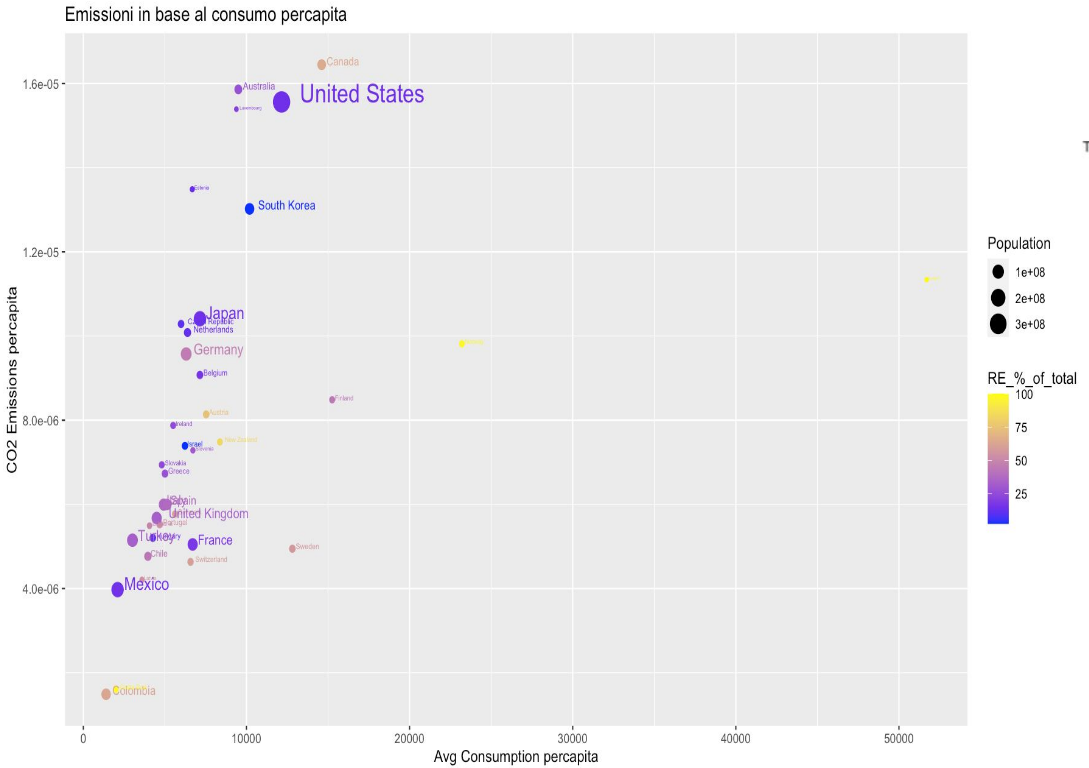
Economy

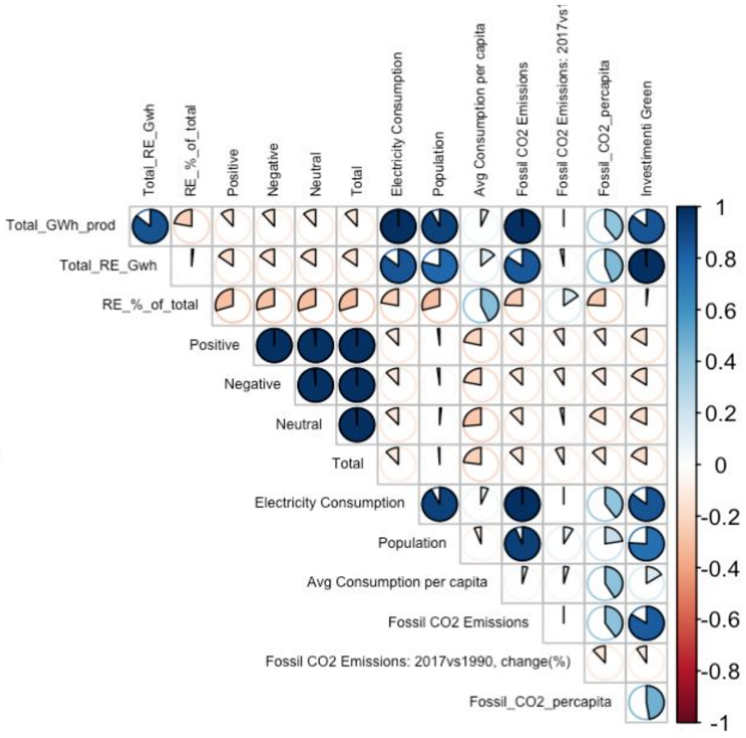
- Green Investment
- Official Development Assistance (Oda)
- Population

Pollution

- Fossil Co2 emissions
- Growth Co2 emissions
- Per capita Co2 emissions

Exploratory Data Analysis





Classification

- Logistic Regression
- KNN
- Random forest
- SVM

Logistic Regression

Training set

```
Call:
glm(formula = `Risultato Investimento` ~ Population + `Investimenti Green` +
    Total_GWh_prod + Total_RE_Gwh, family = binomial, data = ds1,
    na.action = na.omit)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-5.641e-04  -2.000e-08  -2.000e-08   2.000e-08   5.469e-04

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)          1.241e+00  1.246e+03   0.001    0.999
Population          -1.326e-05  9.427e-04  -0.014    0.989
`Investimenti Green` -1.559e+02  1.266e+04  -0.012    0.990
Total_GWh_prod      -1.030e-02  6.877e-01  -0.015    0.988
Total_RE_Gwh         3.243e-02  2.137e+00   0.015    0.988

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4.6626e+01  on 36  degrees of freedom
Residual deviance: 7.1783e-07  on 32  degrees of freedom
AIC: 10

Number of Fisher Scoring iterations: 25
```

Test set - CV

```
Generalized Linear Model

37 samples
16 predictors
 2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 33, 34, 34, 33, 34, 33, ...
Resampling results:

  Accuracy   Kappa
  0.785      0.5785714
```

Precision, Recall and F1-score

```
precision_glm <- posPredValue(predictions_glm, y_glm, positive="Positive") # 1
recall_glm <- sensitivity(predictions_glm, y_glm, positive="Positive") # 1

F1_glm <- (2 * precision_glm * recall_glm) / (precision_glm + recall_glm) # 1
```

KNN

Training set / Test set

```
p.YTrain3 = knn(XTrain, XTrain, YTrain, k=12)
table(ds1$`Risultato Investimento`,p.YTrain3)
mean(p.YTrain3!=ds1$`Risultato Investimento`) ## ErrorRate = 0.3513514
mean(p.YTrain3==ds1$`Risultato Investimento`) ## Accuracy 0.6486486
```

```
   p.YTrain3
    0  1
0 22  3
1 10  2
```

```
p.YTest3 = knn(XTrain, XTest, YTrain, k=12)
table(ds1[test,]$`Risultato Investimento`,p.YTest3)
mean(p.YTest3!=ds1[test,]$`Risultato Investimento`) #ErrorRate 0.1538462
mean(p.YTest3==ds1[test,]$`Risultato Investimento`) #Accuracy 0.8461538
```

```
   p.YTest3
    0  1
0 10  0
1  2  1
```

Precision, Recall and F1-score

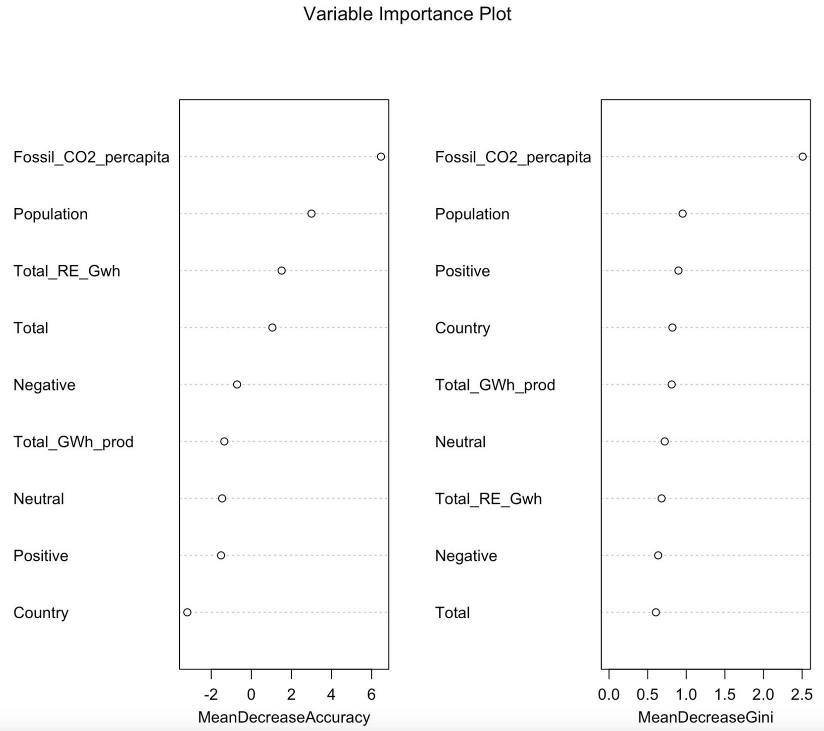
```
precision <- posPredValue(predictions, y, positive="Positive") # 0.4545455
recall <- sensitivity(predictions, y, positive="Positive") # 0.4166667
F1 <- (2 * precision * recall) / (precision + recall) # 0.4347826
```

Random Forest

Training set / Test set

```
rf2=randomForest(ds1_train$`Risultato Investimento`~.,
                 data=ds1_train[,-c(4,9,11:16,19)],
                 mtry=4,importance=TRUE,ntree=500, na.action = na.omit)
summary(rf2)
rf_preds2 = predict(rf2,newdata=ds1_test)
rf_preds2
table(rf_preds2, ds1_test$`Risultato Investimento`)
8/13 ## Accuracy = 0.6153846
1-(8/13) ## ErrorRate = 0.3846154
```

```
rf_preds2 0 1
        0 7 5
        1 0 1
```



Precision, Recall and F1-score

```
precision_rf <- posPredValue(predictions_rf, y_rf, positive="1") #  1
recall_rf <- sensitivity(predictions_rf, y_rf, positive="1") # 0.1666667
F1_rf <- (2 * precision_rf * recall_rf) / (precision_rf + recall_rf) # 0.2857143
```

SVM

1st SVM

```
mean(pred_test1==ds_svm_test$`Risultato Investimento`) #accuracy 0.8461538
precision_svm <- posPredValue(predictions_svm, y_svm, positive="1") # 0.6666667
recall_svm <- sensitivity(predictions_svm, y_svm, positive="1") # 0.6666667
F1_svm <- (2 * precision_svm * recall_svm) / (precision_svm + recall_svm) #0.6666667
```

```
       pred
truth -1 1
   -1  9 1
   1   1 2
```

2nd SVM

```
pred_test2 <-predict(svm_model2,ds_svm_test)
table("truth"=ds_svm_test$`Risultato Investimento`, "pred"=pred_test2)
mean(pred_test2==ds_svm_test$`Risultato Investimento`) #[1] 0.8181818
mean(pred_test2!=ds_svm_test$`Risultato Investimento`) #[1] 0.1818182 tasso errata classificazione test
```

Precision, Recall and F1-score

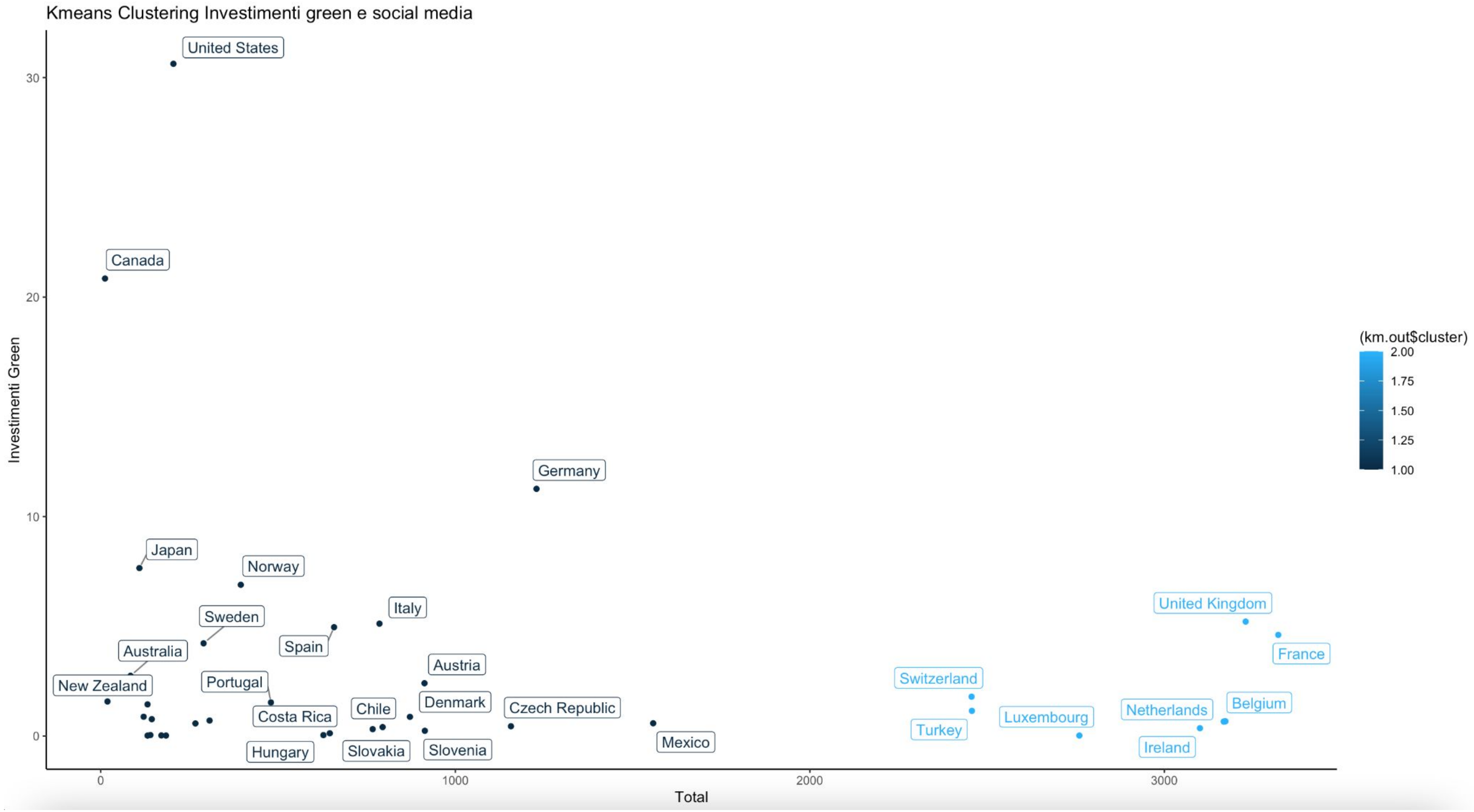
```
precision_svm <- posPredValue(predictions_svm, y_svm, positive="1") # 0.6666667
recall_svm <- sensitivity(predictions_svm, y_svm, positive="1") # 0.6666667

F1_svm <- (2 * precision_svm * recall_svm) / (precision_svm + recall_svm) # 0.6666667
```

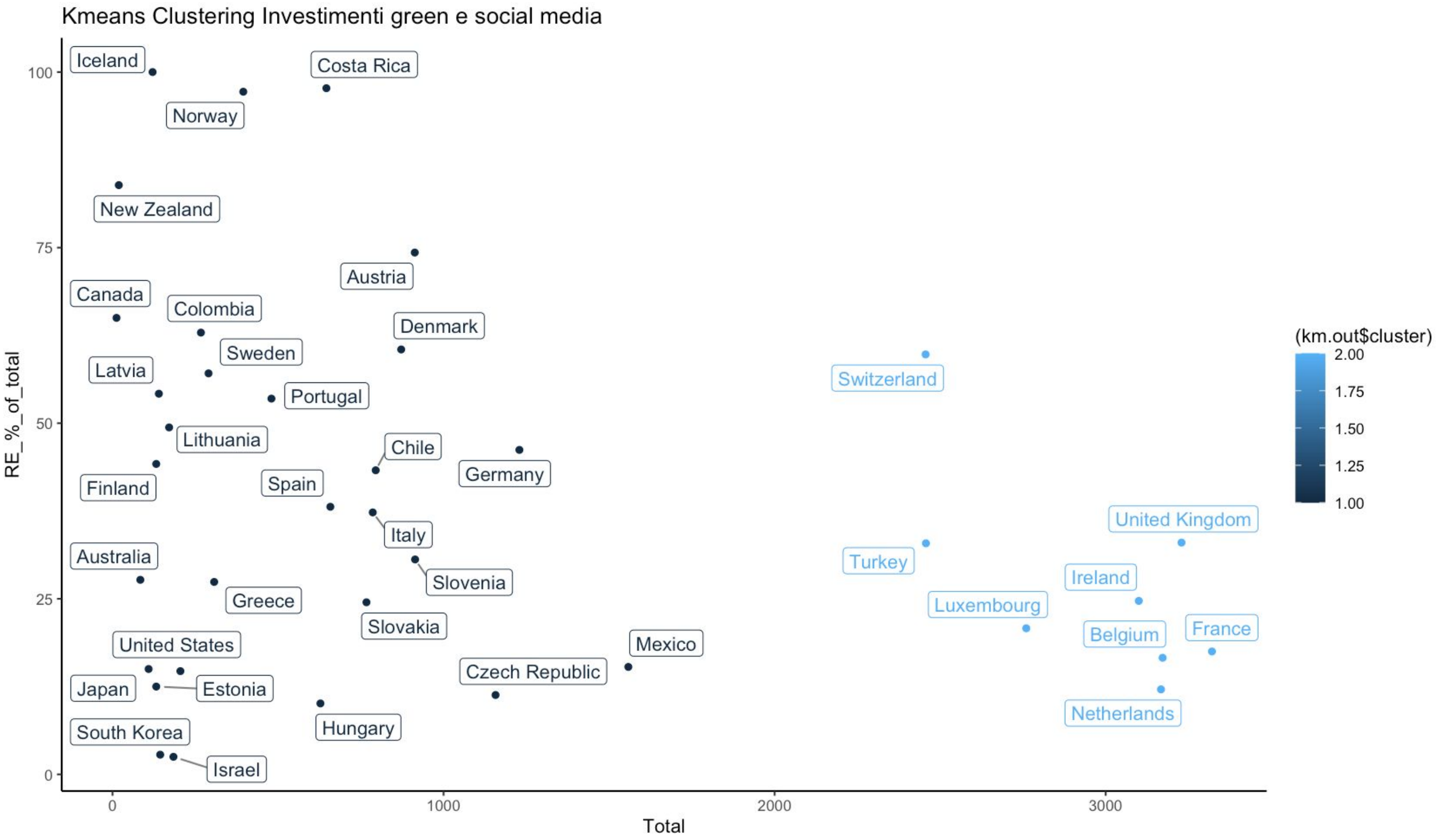
Cluster Analysis

- K-means Clustering
- Hierarchical Clustering
- Principal Component Analysis
- Clustering with PCA

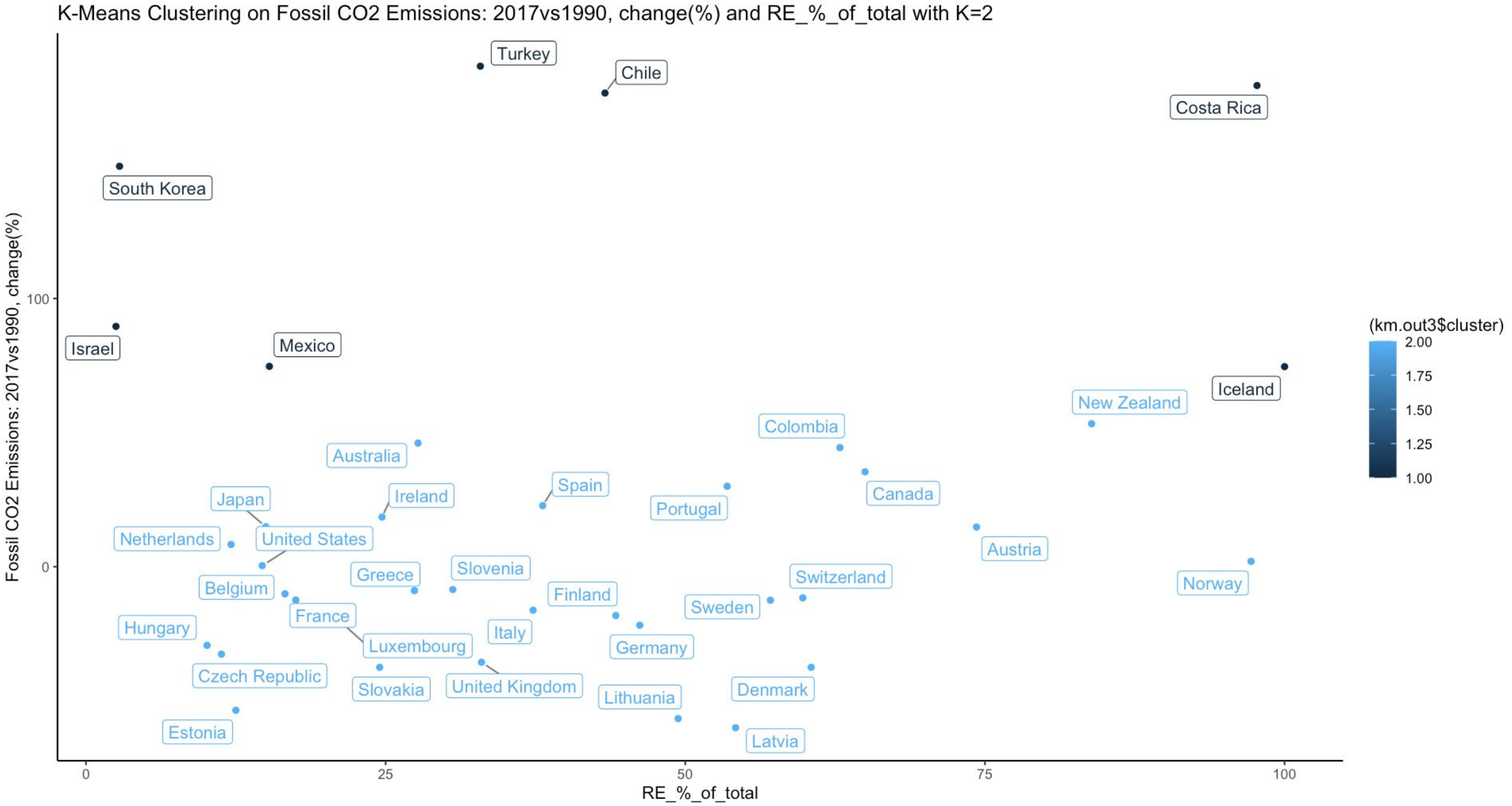
K-Means Clustering - 1



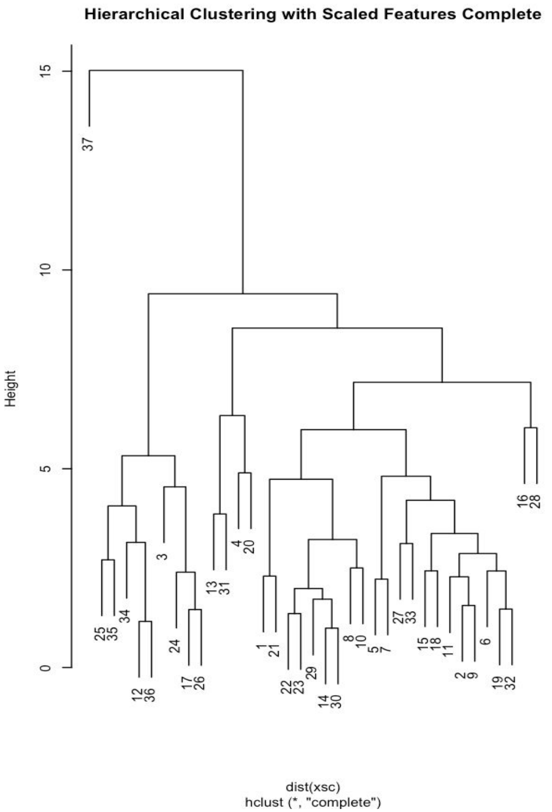
K-Means Clustering - 2

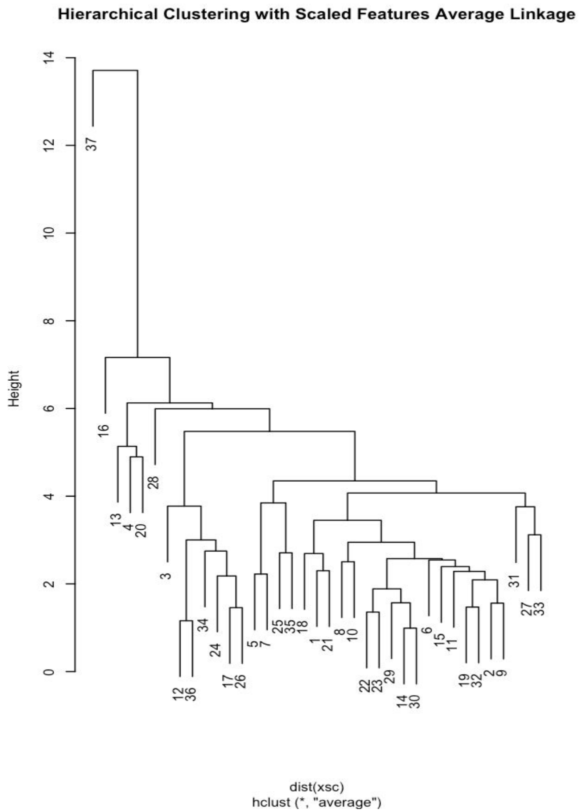


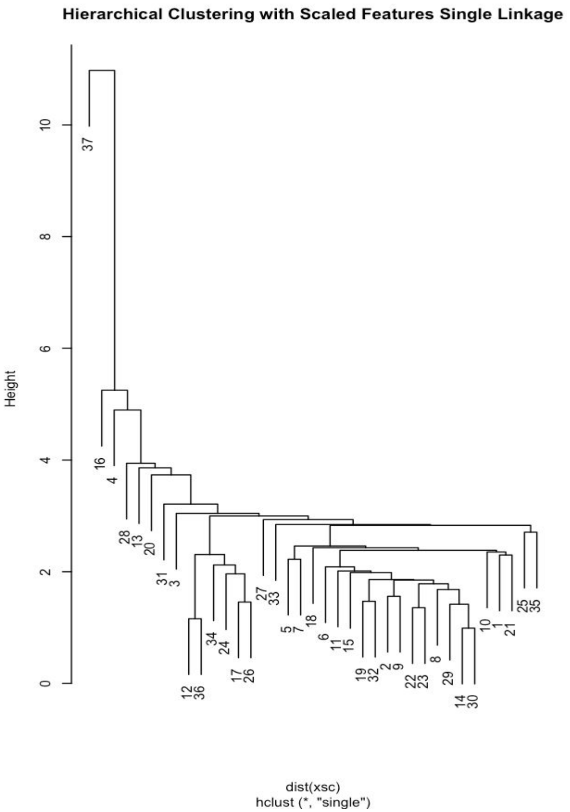
K-Means Clustering - 3



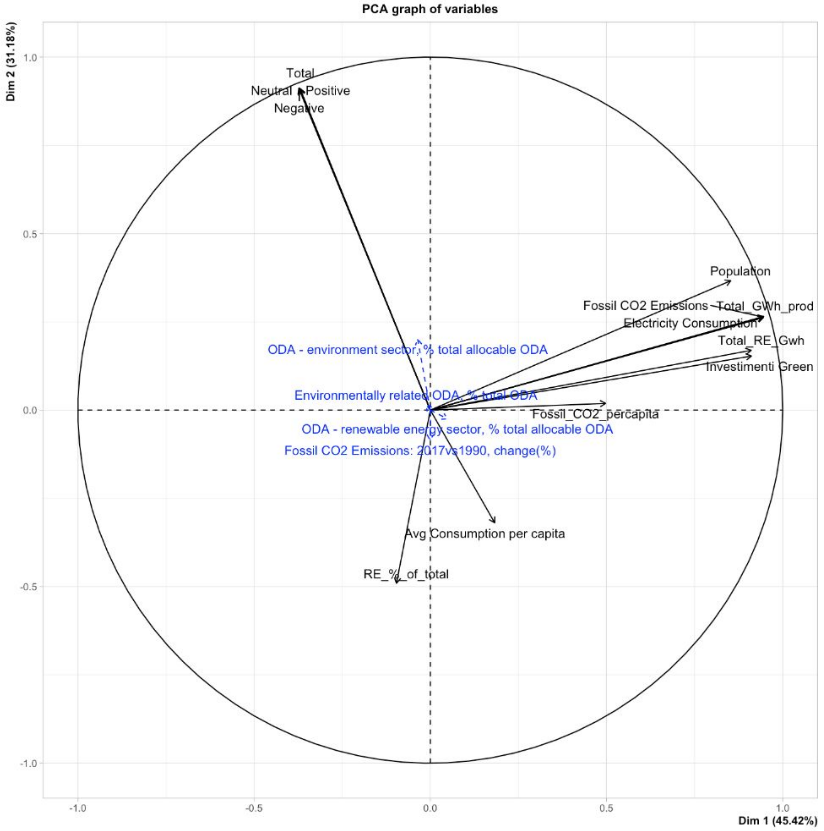
Hierarchical Clustering







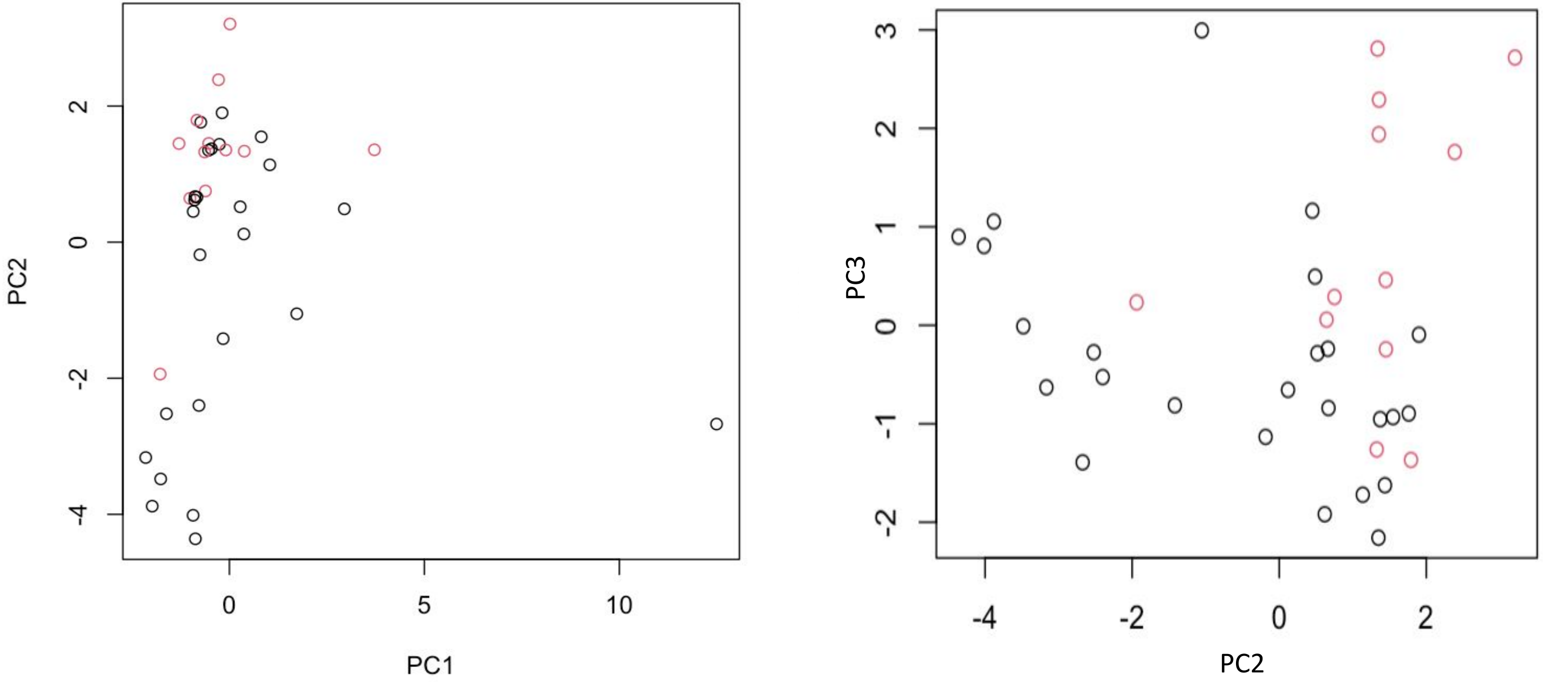
Principal Component Analysis - 1



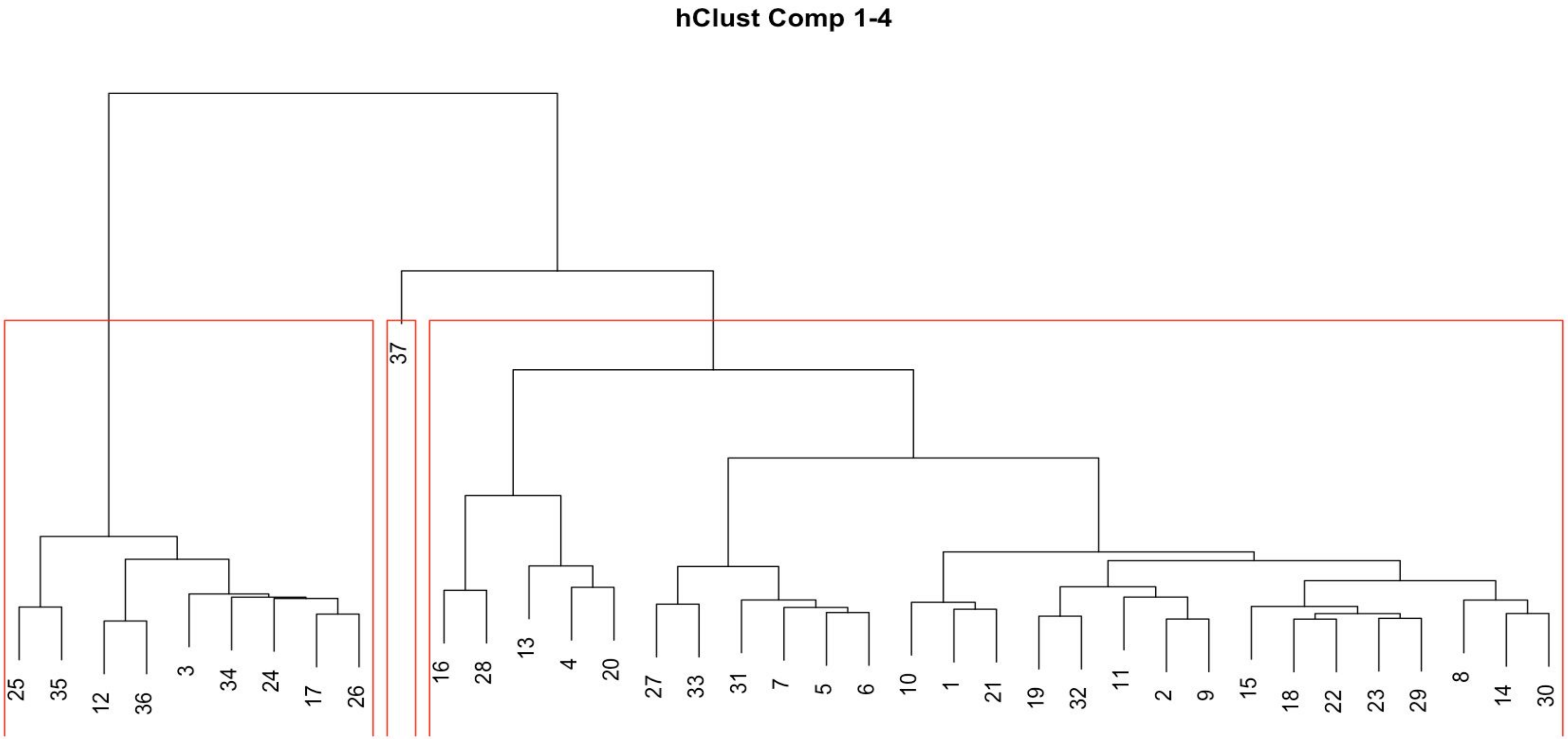

```
#Eigenvalues
#                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5
#Variance                5.905   4.053   1.350   1.015   0.433
#% of var.               45.424  31.180  10.386   7.810
3.331
#Cumulative % of var.     45.424  76.603  86.990  94.799
98.130

                         Dim.5   Dim.6   Dim.7   Dim.8   Dim.9
#Variance                0.433   0.149   0.077   0.009   0.003
#% of var.               3.331   1.145   0.593   0.070   0.026
#Cumulative % of var.    98.130  99.275  99.868  99.938
99.963
```

Principal Component Analysis - 2



Clustering with PCA



Conclusions