Replication of RQ1 on claustering analyses from the paper

Load input file

We load the file with the scores from LIWC and rescale them in the range [1,5].

```
full_personality_df = read_delim(params$data, ";", escape_double = FALSE)
full_personality_df$openness <- resca(full_personality_df,openness,
                                      new_min=1, new_max=5)$openness_res
full_personality_df$conscientiousness <- resca(full_personality_df,conscientiousness,
                                               new_min=1, new_max=5)$conscientiousness_res
full_personality_df$extraversion <- resca(full_personality_df,extraversion,
                                          new_min=1, new_max=5)$extraversion_res
full_personality_df$agreeableness <- resca(full_personality_df,agreeableness,
                                           new_min=1, new_max=5)$agreeableness_res
full_personality_df$neuroticism <- resca(full_personality_df,neuroticism,
                                         new_min=1, new_max=5)$neuroticism_res
```

Now we drop the unnecessary columns and, for each trait, we compute the average score per developer. An overview of the data just loaded:

```
query = sqldf::sqldf(
  "select uid, avg(openness) as 'openness',
    avg(conscientiousness) as 'conscientiousness',
    avg(extraversion) as 'extraversion',
    avg(agreeableness) as 'agreeableness',
    avg(neuroticism) as 'neuroticism'
  from `full_personality_df` group by uid"
)
personality <-
  dplyr::select(
    query,
    openness,
    conscientiousness,
    extraversion,
    agreeableness,
    neuroticism
  )

head(personality)
```

```
##   openness conscientiousness extraversion agreeableness neuroticism
## 1     3.83              4.04         2.04          2.81        2.31
## 2     3.89              3.96         1.90          2.61        2.34
## 3     4.07              3.96         1.93          2.57        2.25
## 4     3.96              4.01         1.99          2.75        2.20
## 5     3.94              3.99         2.06          2.74        2.20
## 6     4.12              3.99         1.86          2.52        2.12
```

Preliminary assessment

We first check if the traits distributions are normally distributed with the Shapiro-Wilk test. Because the p-values for all five tests are < 0.05, the data for all the traits significantly deviate from a normal distribution. Hence, as in the original study, we will use non-parametric tests, which do not assume normality in the distribution of data.

```
shapiro.test(personality$openness)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  personality$openness
## W = 1, p-value = 1e-06
```

```
shapiro.test(personality$conscientiousness)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  personality$conscientiousness
## W = 0.9, p-value = 7e-13
```

```
shapiro.test(personality$extraversion)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  personality$extraversion
## W = 0.9, p-value = 9e-11
```

```
shapiro.test(personality$agreeableness)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  personality$agreeableness
## W = 0.9, p-value = 4e-10
```

```
shapiro.test(personality$neuroticism)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  personality$neuroticism
## W = 0.9, p-value = 7e-13
```

In addition, we perform a couple of tests to assess the suitability of our data for structure detection. To ensure that there is a sufficient proportion of variance in our variables that might be caused by underlying factors, we compute the Kaiser-Meyer-Olkin measure, which is equal to 0.5, that is, the minimum acceptable value as suggested in literature; then, we perform Barlett's test of sphericity, which is significant (chi-square=900, $p<0.001$). These results suggest that our data is suitable for structure detection.

```
# Kaiser, Meyer, Olkin Measure of Sampling Adequacy (0.5 is the minimum)
round(KMO(personality)$MSA, 1)
```

```
## [1] 0.5
```

```
# Barlett's test of sphericity
cortest.bartlett(personality)
```

```
## R was not square, finding R from data

## $chisq
## [1] 900
##
## $p.value
## [1] 6.21e-187
##
## $df
## [1] 10
```

Finally, to rule out changes in personality over time, we split the dataset by date into two sections. Specifically, for each developers, we assess the time-span between the first and last communication in the dataset; then, we compute the point in time Mt so that approximately half of the observations (i.e., the monthly-based personality scores) are located before and after it. Then, two aggregate profiles for each developer are created by averaging the trait scores. Finally, for each trait, we perform a Wilcoxon Signed-Rank test to verify the null hypothesis that the median difference between pairs of observations (i.e., for each subject) is not significantly different from zero. Table 5 reports the results from the five paired tests, which show no significant differences between the distributions (all adjusted p-values > 0.05 after Bonferroni correction for multiple tests), thus confirming the stability of personality traits over time.

```
ids =  sqldf::sqldf("SELECT DISTINCT uid,
                    count(uid) as obs
                    FROM full_personality_df
                    GROUP by uid
                    ORDER BY uid ASC;")
mt.vector = sqldf::sqldf("SELECT uid,
                         email_count as emails,
                         word_count as words
                         FROM full_personality_df
                         ORDER BY uid ASC;")

before = data.frame(matrix(nrow=0, ncol=6))
colnames(before) <- c("uid", "O", "C", "E", "A", "N")
after = data.frame(matrix(nrow=0, ncol=6))
colnames(after) <- c("uid", "O", "C", "E", "A", "N")
for(i in 1:nrow(ids)) {
  id1 = ids[i, 1]
  obs = ids [i, 2]
  spl1 = round(obs/2, 0)

  if(spl1 > 1) { # correction, avoid unpaired values and ties

    a =  sqldf::sqldf(sprintf("SELECT uid,
                              openness AS O,
                              conscientiousness AS C,
                              extraversion AS E,
                              agreeableness AS A,
                              neuroticism AS N
                              FROM full_personality_df
                              where uid = '%s'
                              order by month DESC
                              limit '%s';", id1, obs-spl1 ))
    x = data.frame(id1,mean(a$O), mean(a$C), mean(a$E), mean(a$A), mean(a$N))
    colnames(x) <- c("uid", "O", "C", "E", "A", "N")
```

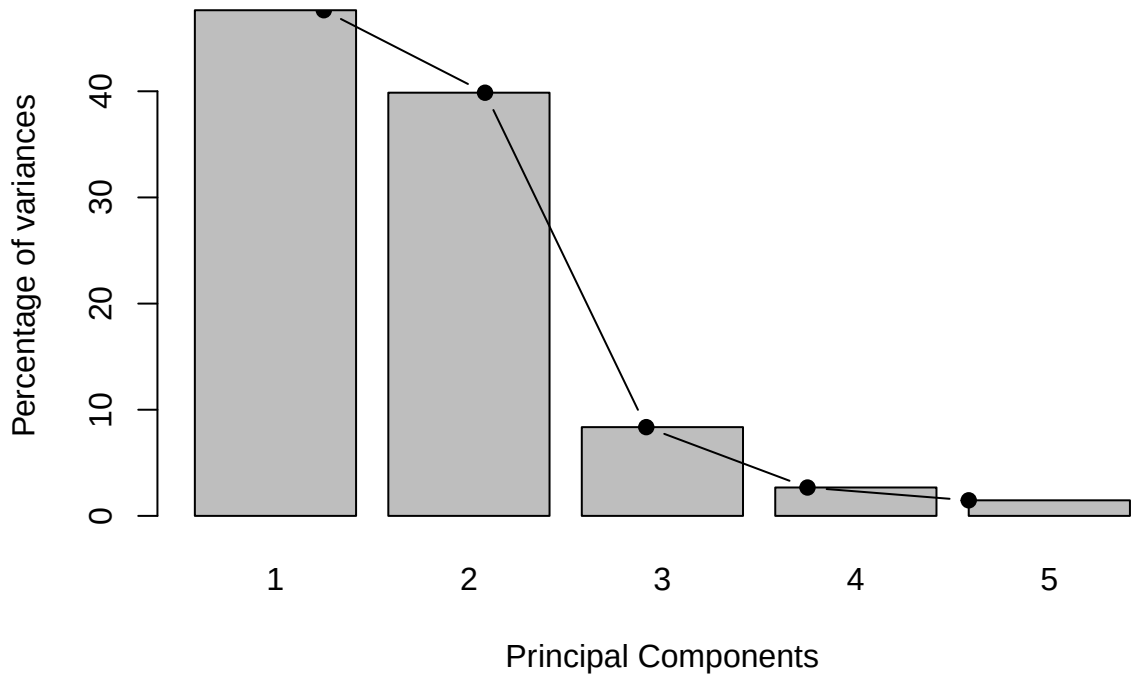
```
    after = rbind(after, x)

    b =  sqldf::sqldf(sprintf("SELECT uid,
                              openness AS O,
                              conscientiousness AS C,
                              extraversion AS E,
                              agreeableness AS A,
                              neuroticism AS N
                              FROM full_personality_df
                              where uid = '%s'
                              order by month ASC
                              limit '%s';", id1, spl1 ))
    x = data.frame(id1,mean(b$O), mean(b$C), mean(b$E), mean(b$A), mean(b$N))
    colnames(x) <- c("uid", "O", "C", "E", "A", "N")
    before = rbind(before, x)
  }
}

wO = wilcox.test(before$O,after$O, paired = TRUE,
                 alternative = "two.sided",exact=TRUE,conf.int=TRUE)
wC = wilcox.test(before$C,after$C, paired = TRUE,
                 alternative = "two.sided",exact=TRUE,conf.int=TRUE)
wE = wilcox.test(before$E,after$E, paired = TRUE,
                 alternative = "two.sided",exact=TRUE,conf.int=TRUE)
wA = wilcox.test(before$A,after$A, paired = TRUE,
                 alternative = "two.sided",exact=TRUE,conf.int=TRUE)
wN = wilcox.test(before$N,after$N, paired = TRUE,
                 alternative = "two.sided",exact=TRUE,conf.int=TRUE)

# bonferroni adjustment
ps <- p.adjust(c(wO$p.value, wC$p.value, wE$p.value, wA$p.value, wN$p.value),
               method = "bonferroni", n=5)
ps <- round(ps, 3)

dfW <- data.frame(matrix(ncol = 5, nrow = 0))
x <- c("Trait", "V", "p-value ", "CI 95% low", "CI 95% high")
colnames(dfW) <- x
dfW[1,] <- c("Openness", as.numeric(wO$statistic), ps[1], round(as.numeric(wO$conf.int), 3))
dfW[2,] <- c("Conscientiousness", as.numeric(wC$statistic), ps[2], round(as.numeric(wC$conf.int), 3))
dfW[3,] <- c("Extraversion", as.numeric(wE$statistic), ps[3], round(as.numeric(wE$conf.int), 3))
dfW[4,] <- c("Agreeableness", as.numeric(wA$statistic), ps[4], round(as.numeric(wA$conf.int), 3))
dfW[5,] <- c("Neuroticism", as.numeric(wN$statistic), ps[5], round(as.numeric(wN$conf.int), 3))
dfW
```

```
##               Trait    V p-value  CI 95% low CI 95% high
## 1          Openness 9330       1     -0.006       0.029
## 2 Conscientiousness 8320       1     -0.014       0.014
## 3      Extraversion 7751       1     -0.022       0.008
## 4     Agreeableness 7199   0.448     -0.028       0.002
## 5       Neuroticism 9075       1     -0.008       0.023
```

Factor analysis with PCA

We perform Principal Component Analysis (PCA) with varimax rotation. PCA is a statistical procedure that converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables, i.e., the principal components. The scree plot below suggest with the elbow method that we can extract either two or three components.

```
personality_log <- personality
res.pca <- FactoMineR::PCA(personality_log, graph = FALSE)
eigenvalues <- res.pca$eig
barplot(eigenvalues[, 2], names.arg=1:nrow(eigenvalues),
        xlab = "Principal Components",
        ylab = "Percentage of variances",
        col ="grey")
# Add connected line segments to the plot
lines(x = 1:nrow(eigenvalues), eigenvalues[, 2],
      type="b", pch=19, col = "black")
```



The analysis of the cumulative proportion of variance shows that the three components that account for 96% of the total variance in the data.

```
fit <- princomp(scale(personality_log,  center = TRUE, scale = TRUE), cor = TRUE)
summary(fit) # print variance accounted for
```

```
## Importance of components:
##                           Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## Standard deviation       1.543  1.412 0.6466 0.3657 0.2709
## Proportion of Variance   0.476  0.399 0.0836 0.0267 0.0147
## Cumulative Proportion    0.476  0.875 0.9586 0.9853 1.0000
```

We complement the screeplot with the analysis of the eigenvalues. The table below shows that only the first two have a value over Kaiser's criterion of 1, the cut-off point typically used to retain principal components. Eigenvalues, in fact, correspond to the amount of the variation explained by each principal component. A component with an eigenvalue > 1 indicates that it accounts for more variance than its accounted by one of the original variables in the dataset.

```
head(round(eigenvalues[, 1:2], 4))
```

```
##        eigenvalue percentage of variance
## comp 1     2.3816                 47.63
## comp 2     1.9932                 39.86
## comp 3     0.4180                  8.36
## comp 4     0.1337                  2.67
## comp 5     0.0734                  1.47
```

```
edf <- as.data.frame(eigenvalues)
ec <- length(edf[edf$eigenvalue>1, 1]) # cutoff eigenvalues > 1.0 to extract components
```

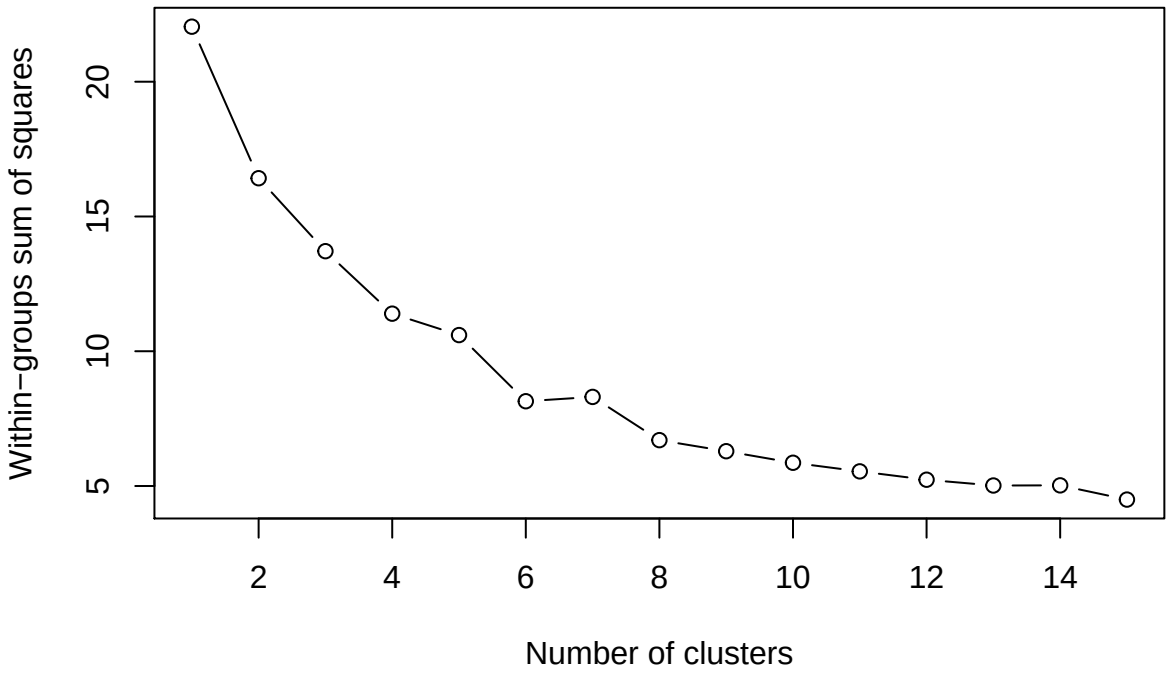
Finally, we show the standardized loadings of the five traits on the two principal components. *Agreebleness*, *extraversion*, and *openness* load on the first component, albeit *openness* loading is negative (hence, lack thereof); instead, *conscientiousness* and *neuroticism* load on the second component, but *neuroticism* loading is negative (hence, it indicates emotional stability).

```
principal(personality_log, nfactors=ec, rotate="varimax")$loadings
```

```
##
## Loadings:
##                   RC1    RC2
## openness          -0.861  0.414
## conscientiousness  0.118  0.942
## extraversion       0.849  0.132
## agreeableness      0.867  0.357
## neuroticism              -0.971
##
##                  RC1   RC2
## SS loadings    2.227 2.148
## Proportion Var 0.445 0.430
## Cumulative Var 0.445 0.875
```

Cluster analysis

As PCA is not the only approach followed in literature to group individuals by similar personality profiles, we apply the k-means clustering algorithm. We use the 'elbow' method to identify the optimal number of cluster from the plot below. The 'elbow' point corresponds to the smallest k value (2 in our case, rather than 6) after which we do not observe a large decrease in the within-group heterogeneity, here measured using the sum of squares, with the increase of the number of clusters.



The table below shows the size of the two clusters obtained with k-means. Although the first cluster is twice the size of the second, using other k values returns even more imbalance clusters. The table also reports the coordinates of the centroids, that is the average position of the elements assigned to a cluster. All the values are z-score standardized, with positive (negative) values above (below) the overall means.

```
K <- 2  # elbow
myclusters <- kmeans(scale(personality_log, center=TRUE, scale=TRUE), K)
myclusters$size
```

```
## [1]  76 156
```

```
round(myclusters$centers, 2)
```

```
##   openness conscientiousness extraversion agreeableness neuroticism
## 1    -1.03            -0.33         0.77          0.65        0.45
## 2     0.50             0.16        -0.38         -0.32       -0.22
```

Because the data are not normally distributed, we perform five nonparametric Kruskal-Wallis (KW) tests to make unpaired comparisons between the two independent score distributions (i.e., the clusters) for each of the five traits. The table below shows the results of the KW tests, after applying Bonferroni corrections of p-values for repeated tests. Each p-value is smaller than 0.001, however the epsilon-squared statistic shows a strong effect size (> 0.36) for *openness* and *extraversion*, a relatively strong effect (> 0.16) for *agreeableness* and *neuroticism*, and a moderate effect size (> 0.04) for *conscientiousness*. Hence, we conclude that there are significant differences among the two clusters.

```
traits <- c("openness", "conscientiousness", "extraversion", "agreeableness", "neuroticism")
dfs <- list()
k <- 1
for (i in 1:K) {
  for (j in 1:length(traits)) {
    assign("trait", traits[j])
    c_i_j <- dplyr::select(personality_log[myclusters$cluster == i, ], y=trait)
    c_i_j$trait <- traits[j]
    c_i_j$cluster <- paste("Cluster", i)
    dfs[[k]] <- c_i_j
```

```
    k <- k + 1
  }
}

df <- do.call(rbind, dfs)
# multiple pairwaise comparison between traits in the clusters with Bonferroni correction
for (i in 1:length(traits)) {
  print("**************************************************************************")
  print(traits[i])
  print("**************************************************************************")
  d = df[df$trait == traits[i], ]
  d$cluster <- as.factor(d$cluster)
  kwt <- kruskal.test(d$y, d$cluster)
  print(kwt)
  print("Corrected p-value (bonferroni)")
  out<-p.adjust(kwt$p.value, method = "bonferroni", n=length(traits))
  print(out)
  eps <- rcompanion::epsilonSquared(d$y, d$cluster, ci=TRUE, conf = 0.95)
  print("effect size")
  print(as.matrix(eps))
}
```

```
## [1] "**************************************************************************"
## [1] "openness"
## [1] "**************************************************************************"
##
##  Kruskal-Wallis rank sum test
##
## data:  d$y and d$cluster
## Kruskal-Wallis chi-squared = 136, df = 1, p-value <2e-16
##
## [1] "Corrected p-value (bonferroni)"
## [1] 8.68e-31
## [1] "effect size"
##      epsilon.squared lower.ci upper.ci
## [1,]            0.59    0.515    0.653
## [1] "**************************************************************************"
## [1] "conscientiousness"
## [1] "**************************************************************************"
##
##  Kruskal-Wallis rank sum test
##
## data:  d$y and d$cluster
## Kruskal-Wallis chi-squared = 17, df = 1, p-value = 4e-05
##
## [1] "Corrected p-value (bonferroni)"
## [1] 0.000201
## [1] "effect size"
##      epsilon.squared lower.ci upper.ci
## [1,]           0.073   0.0212    0.147
## [1] "**************************************************************************"
## [1] "extraversion"
## [1] "**************************************************************************"
##
```

```
##  Kruskal-Wallis rank sum test
##
## data:  d$y and d$cluster
## Kruskal-Wallis chi-squared = 84, df = 1, p-value <2e-16
##
## [1] "Corrected p-value (bonferroni)"
## [1] 3.14e-19
## [1] "effect size"
##      epsilon.squared lower.ci upper.ci
## [1,]           0.362    0.248    0.465
## [1] "**********************************************************************"
## [1] "agreeableness"
## [1] "**********************************************************************"
##
##  Kruskal-Wallis rank sum test
##
## data:  d$y and d$cluster
## Kruskal-Wallis chi-squared = 62, df = 1, p-value = 3e-15
##
## [1] "Corrected p-value (bonferroni)"
## [1] 1.38e-14
## [1] "effect size"
##      epsilon.squared lower.ci upper.ci
## [1,]            0.27    0.169    0.375
## [1] "**********************************************************************"
## [1] "neuroticism"
## [1] "**********************************************************************"
##
##  Kruskal-Wallis rank sum test
##
## data:  d$y and d$cluster
## Kruskal-Wallis chi-squared = 24, df = 1, p-value = 1e-06
##
## [1] "Corrected p-value (bonferroni)"
## [1] 4.79e-06
## [1] "effect size"
##      epsilon.squared lower.ci upper.ci
## [1,]           0.104   0.0404    0.184
```

```r
# threshold for epsilonSquared interpretation from
# Rea, L. M., & Parker, R. A. (1992). Designing and conducting survey research:
# a comprehensive guide.

# 0.00 < 0.01 - Negligible
# 0.01 < 0.04 - Weak
# 0.04 < 0.16 - Moderate
# 0.16 < 0.36 - Relatively strong
# 0.36 < 0.64 - Strong
# 0.64 < 1.00 - Very strong
```

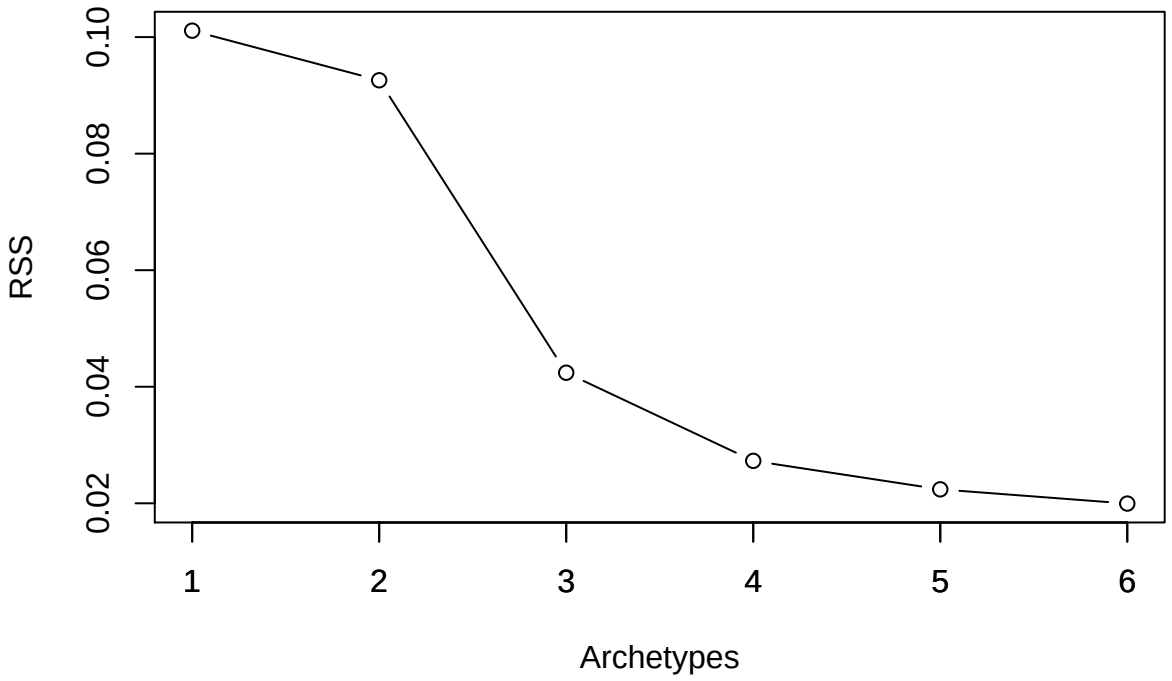
Finally, by comparing the traits values across the two clusters, we identify two opposite clusters. Accordingly, we label Cluster 1 as the subgroup of the 'close-minded, impulsive, outgoing, warm, and emotionally unstable,' since on average they score lower in *openness* and *conscientiousness*, and higher in *extraversion*, *agreeableness*, and *neuroticism*. Cluster 2 is the opposite subgroup of developers who are more 'open to experience, dependable, solitary, cold, and stable,' given that they exhibit higher average scores in *openness*

and *conscientiousness*, and lower scores in *extraversion*, *agreeableness*, and *neuroticism*.

Archetypal analsysis

Finally, we perform Archetypal Analysis to extract personality groupings. We use the 'elbow' criterion again to identify the optimal number of archetypes to extract. From the scree plot below, which shows the fraction of total variance in the data explained by the number of extracted archetypes, we notice that the function plateaus after extracting 2 or 3 archetypes. For the sake of simplicity in characterizing the archetypes, we opt for extracting 2.

```
screeplot(arc)
```




```
arc_best <- bestModel(arc[[3]])
```

Table 10 shows the trait coordinates for both archetypes, standardized for the ease of comparison. We compare the trait values across the three archetypes and obtain results in line with the findings from k-means. In fact, the extracted archetypes can be mapped on the two clusters described above, since we find that Archetype 1 is similar to Cluster 1, grouping developers scoring lower in *openness* and *conscientiousness*, and higher in *extraversion*, *agreeableness*, and *neuroticism*; Archetype 2 is similar to Cluster 2, grouping developers with higher scores in *openness* and *conscientiousness*, and lower scores in *extraversion*, *agreeableness*, and *neuroticism*.

```
scale(parameters(arc_best)[1,], center=TRUE, scale=TRUE)
```

```
##                     [,1]
## openness           1.169
## conscientiousness  0.705
## extraversion      -1.284
## agreeableness     -0.691
## neuroticism        0.102
## attr(,"scaled:center")
## [1] 2.8
## attr(,"scaled:scale")
## [1] 0.831
```

```
scale(parameters(arc_best)[2,], center=TRUE, scale=TRUE)
```

```
##                      [,1]
## openness            1.096
## conscientiousness   1.025
## extraversion       -0.954
## agreeableness      -0.305
## neuroticism        -0.862
## attr(,"scaled:center")
## [1] 2.97
## attr(,"scaled:scale")
## [1] 1.2
```

```
scale(parameters(arc_best)[3,], center=TRUE, scale=TRUE)
```

```
##                      [,1]
## openness            0.389
## conscientiousness   1.288
## extraversion       -0.726
## agreeableness       0.303
## neuroticism        -1.253
## attr(,"scaled:center")
## [1] 3.11
## attr(,"scaled:scale")
## [1] 0.837
```