Statistical learning - individual assignment 1

Joshua Damm

2024-04-22

Part A: Supervised

Question 1

When looking at the relevant predictor variables X_1-X_3, we can see that that they are drawn from uniform and normal distributions. All noise variables are drawn from uniform distributions. The response is set up as a non-linear transformation of the three predictor variables, including polynomials and indicator variables, suggesting a complex non-linear relationship between the predictors and the response.

Considering the bias-variance trade-off and the 0-1 loss framework, KNN tends to have lower bias but higher variance compared to parametric models like LASSO logistic regression. In situations where the underlying relationship is complex or non-linear, KNN might perform better due to its flexibility, while LASSO might generalize better if the true relationship is simpler and can be captured by a linear model. Therefore, for the variables X_1-X_6, kNN might be able to detect potentially non-linear relationships. However, kNN is also sensitive to noise and overfitting especially when irrelevant predictors like X_4-X_6 are included. LASSO logistic regression, known for its regularization strength, introduces bias through coefficient shrinkage but significantly lowers variance by effectively eliminating irrelevant predictors. This attribute of LASSO, where it zeroes out coefficients for noise variables, simplifies the model and focuses it on the relevant variables, thereby enhancing predictive accuracy and robustness.

Given the setup with three relevant predictors and three noise variables, LASSO might shrink the predictors of the noise variables to low values or even to zero, but might also fail to detect the non-linear relationsships in the data. kNN has a tendency to overfit, but given only three noise variables, its decision boundary might still provide a good fit for classification and it is able to detect non-linearities. Therefore, I expect kNN to outperform LLR.

Question 2

In this scenario when looking at all predictors, i.e. three relevant predictor variables X_1-X_3 and 200 noise variables X_4 -X_{203} I expect the property of Lasso Logistic Regression penalization, dimensionality-reduction and variance reduction properties to shine by eliminating a decent amount of noise variables from the final model. The LLR model might still fail to detect the non-linear relationships between the predictors and the response, but it might still give a good enough fit to predict the outcome with a decent accuracy. KNN on the other hand is very likely to overfit due to the massive amount of noise in the data, probably leading to a low prediction accuracy.

Question 3

Question 3a

First I created a subset with the 3 relevant predictors X_1 - X_3 and 3 noise variables X_4 - X_6. Then I split the data into training and test data on a 50/50 split procedure. Then I partitioned the training data into 10 random distinct subsets and ran kNN so that each group once was treated as the test set (10-fold cross validation). This procedure was repeated for every k between 1 and 200 to fit the kNN model. Looking at the prediction accuracy on the training data revealed that a k-value of 79 brings the best classification accuracy of 72,01956%. Given the training set of 5000 observations, this seems to be a reasonable value for k as it provides a good balance between under- and overfitting, potentially slighly underfitting the data, because with k = 79 it smoothes the decision boundary significantly and provides a more generalized result. However, it is crucial to be aware of the fact that KNN is very prone to catching noise in the data, and in this training sample we already deal with 3 noise variables, which might underfit the kNN model. When predicting Y on the test data, the kNN model achieved an accuracy of 71.74%.

Question 3b

First I rearranged the data so that I have a design matrix with variables X_1 - X_6 and a binary response vector Y for both training and testing data. Then I performed Lasso Logistic Regression using 10-fold cross validation to estimate the optimal lambda parameter which is used to introduce a little bias to the logistic regression function in order to reduce the variance and shrink coefficients that are not relevant for the classification of Y. An optimal shrinkage parameter lambda of 0.0103088 was found. For the final model a Lasso Logistic Regression with the optimal lambda was fit on the full training data,leading to a logistic regression equation of

$$\log\left(\frac{p}{1-p}\right) = -0.3262648 + 0.682162745X_1 + 0.131761208X_2 + 0.167683487X_3 + 0.008122535X_4$$

with p being the probability of $Y = 1$. Therefore, the noise variable coefficients for X_5 and X_6 were shrunken down to 0 and for the noise variable X_4 the coefficient is shrunken down to a very low value of appr. 0.008. Finally, this model was used to predict the response Y on the test data, and misclassifications on the true Y values were averaged, showing a classification accuracy of 67.28%.

Question 3c

After fitting both kNN and Lasso Logistic Regression on the training data and testing their accuracy on the test data, kNN revealed a prediction accuracy of 71.74%, wheres Lasso Logistic Regression only achieved an accuracy of 67.28%. The results are somewhat expected, since kNN outperformed Lasso Logistic Regression (LLR) in terms of prediction accuracy, probably due to catching the non-linear relationships between the predictors and the response and by providing a well fitted decision boundary to still distinguish between relevant predictors and noise in the data. LLR on the other hand, was able to shrink all three noise variables X_4 - X_6 close or precisely to zero. This would suggest that LLR would be able to give a higher prediction accuracy on the test data. However, in this case detecting the non-linear relationships might be more important to catch the pattern in the data than separating signal from noise.

Question 4

Question 4a

I ran kNN on the whole data set including all relevant predictors and noise variables. When trained on the training data using 10-fold cross validation k = 41 revealed the best prediction accuracy of 58.62%. The procedure was the same as in 3a except including all variables. This fitted kNN model was then used to predict the reponse in the test data, showing a test accuracy of 57.14%.

Question 4b

I fitted the Lasso Logistic Regression (LLR) model using the same procedure as in 3b, this time including all variables. 10-fold cross validation revealed an optimal lambda tuning parameter of 0.005899074. Importantly, LLR was able to shrink all noise predictors close to or precisely to zero, thereby effectively reducing variance and dimensionality. Fitting the LLR with the optimal lambda again on the full test data and then using this model to predict the outcome Y on the test data revealed a test accuracy of 67.02%.

Question 4c

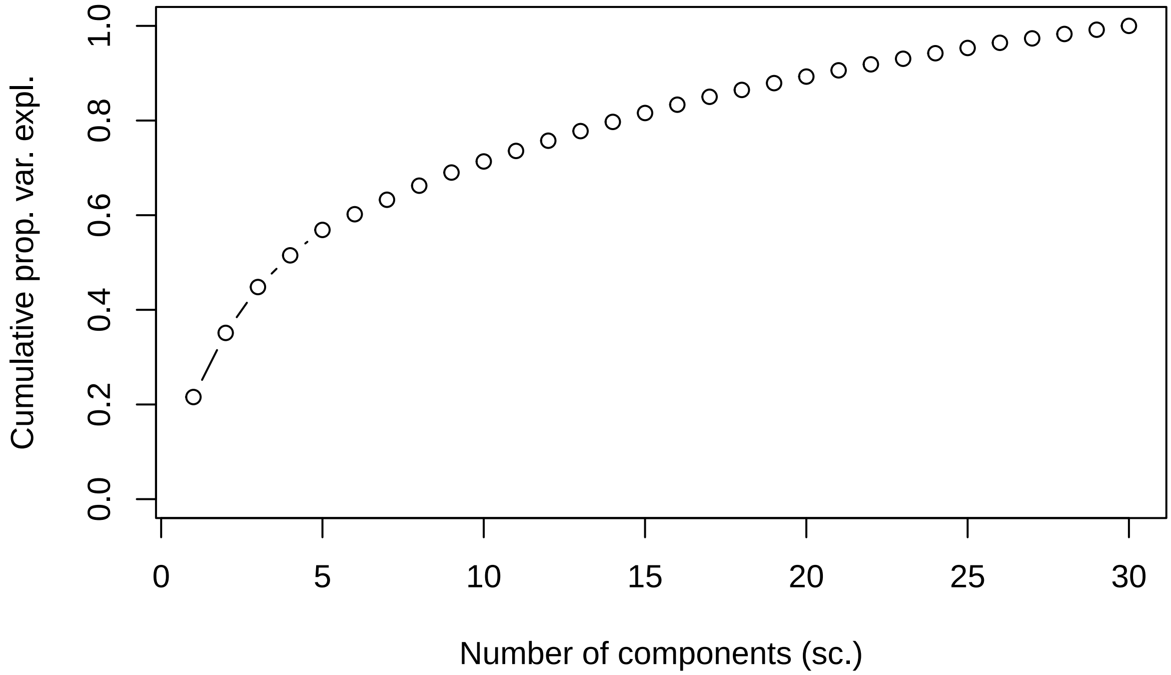
When comparing the accuracy of both models, kNN showed a prediction accuracy of 57.14% on the test data, whereas Lasso Logistic Regression (LLR) revealed an accuracy of 67.02%. This result is expected as in the case of including all noise variables X_4 - X_{203} LLR is expected to shrink these variables close or precisely to zero, which occured in the present analysis. LLR is still not able to detect non-linear relationships between the predictors and the response, but LLR's linear classification was still sufficient to show a solid prediction accuracy. kNN on the other hand, already showed bias when fitted on the training data (62.76%), and showed poorer prediction accuracy on the test data, as it was not able to separate the relevant predictors from the noise variables. This suggests that kNN in this context shows both high variance and high bias.

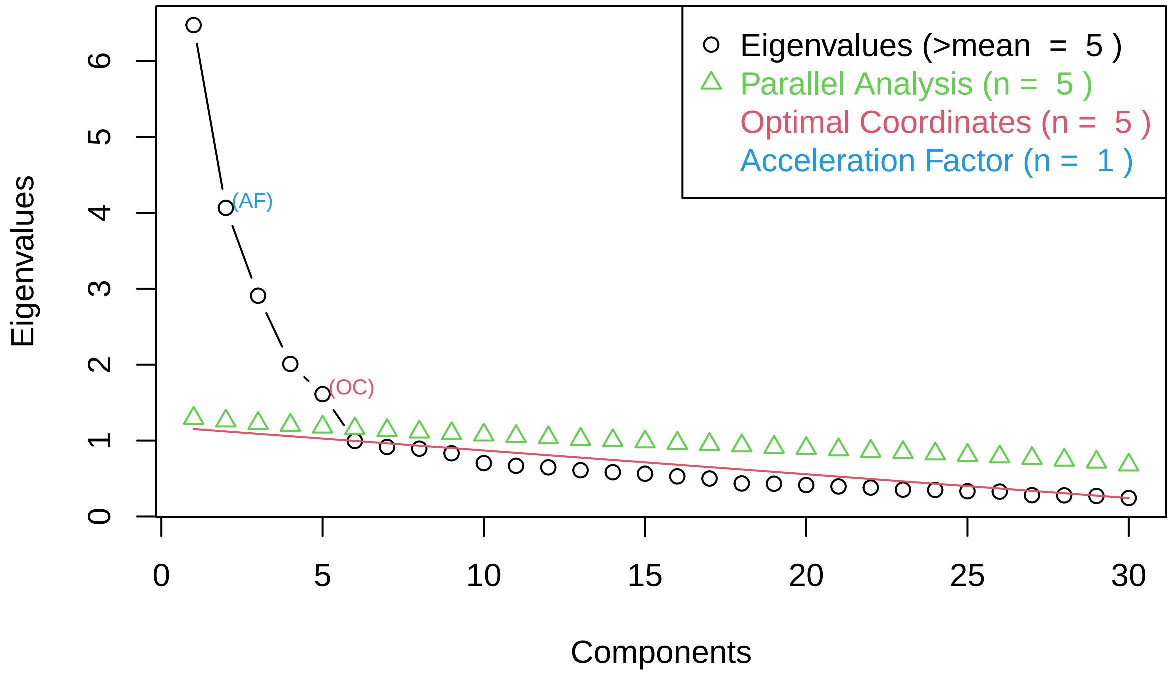
PART B: UNSUPERVISED LEARNING

Question B1

Dimensionality-reduction in this case might be a good idea in case variables are highly correlated with each other. One must be aware though that information might get lost when we reduce dimensionality, possibly leading to a less nuanced assessment of personality groups. On the other hand reducing dimensionality is beneficial to get a clearer sight on which core facets of personality explains the variation in our student sample. Therefore it is easier to understand, describe and interpret the different personality groups eventually. One method to achieve this grouping and dimensionality reduction might be Principal Component Analysis (PCA). PCA transforms the original variables into a new set of variables (principal components), which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables.PCA is appropriate when the goal is to reduce the dimensionality in the data while retaining as much of the variability as possible. It's especially useful when variables are highly correlated like in the present sample, and the goal is to summarize the variations in personality with a smaller number of representative variables that collectively explain most of the variability.

Question B2


Cumulative prop. var. expl.
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
30
Number of components (sc.)



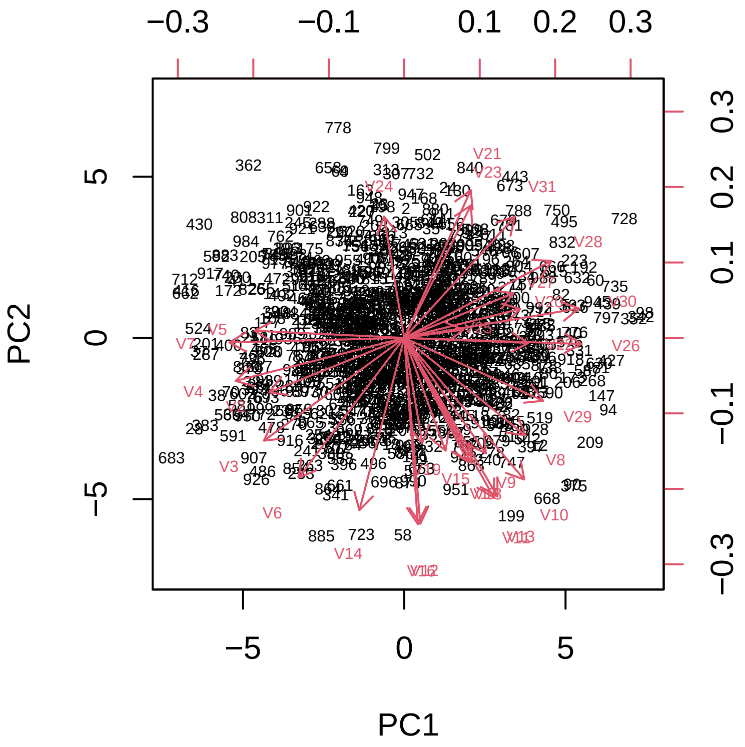
I ran Principal Component Analysis (PCA) on the already scaled variables and decided to use the proportion of variance explained (PVE), cumulative proportion of variance explained (CPVE), a scree-plot (elbow-plot), and Horn's parallel analysis to decide which principal components (PC's) to keep derive new personality variables in order to group the subjects. The scree plot plots the CPVE for each PC in ascending order. The CPVE for a PC is computed as the cumulative sum of the PVE of each component up to the component of interest. As one can see in the scree (elbow) plot, there is a knick in the elbow plot at PC 5. I also looked at the ratios of the change in CPVE between the PC's which suggests that going from component 5 to component 6 (ratio = 1.619536) there the highest drop in CPVE. PC5 has a PVE of 5.37%, whereas PC6 only has a PVE of 3.32%, followed by equally low PVE's for the remaining PC's. So PVE, CPVE and the elbow method suggest to keep 5 components. This is also confirmed by using Kaiser's rule. When looking at the Eigenvalues of the correlation matrix, PC1 - PC5 show an Eigenvalue which is greater than 1. PC6 shows a tendency to reach Kaiser's criterion (lambda = 0.9949349). Runnings Horn's Parralel Analysis shows that the 5th Eigenvalue from our data set is still greater than the mean of the Eigenvalues of the simulated parallel data. However, as one can see on the scree plot for parallel analysis, the 6th Eigenvalue falls right on the mean of the Eigenvalues of the simulated data. In summary, PVE, CPVE, and the elbow method suggests to keep 5 PC's, whereas Kaiser's rule and parallel analysis would also allow to keep 6 components under reasonable scientific circumstances. I decide to be conservative and keep 5 components, since PC6 would only explain 3.32% variance additionally and would complicate interpretability. Altogether, these 5 PC's explain 56.89% of the variance in the data.

Question B3

To give meaning to the new derived variables / Principal Components (PC's) one has to see how the original variables load on the new derived variables. PCA ensures that the new variables capture as much of the variability in the original variables as possible. However, one has to to keep in mind that only 56.89% of the variance in the data could be explained by using our selected 5 PC's. After running PCA, I investigated the loadings of the original variables on the newly found PC's. In order to see patters in the data and interpret the results, I performed varimax rotation, which rotates the axes of the principal components to maximize the variance of the squared loadings of a PC on all the original variables in a given loading matrix. The goal is to make high loadings higher and low loadings lower for each PC, hence simplifying the PC's for interpretability. For a very broad assessment, I could extract some obvious pattern in the rotated loading matrix. For PC1 expecially V1 - to V6 show very high negative loadings on this PC. For PC2, V8 - V13 show very high negative loadings, whereas for PC3, V20 - V25 show high positive loadings. For PC4, V26-V31 show high negative loadings, and for PC5 V14 - V19 show hgh negative loadings. Another obvious observation is that for PC5, only V14 to V19 load positively, one other variable loads slightly negatively (V27), but all other variables do not load at all on this PC after varimax, suggesting that this component captures a unique aspect of personality variation in the data.

For interpretability, PC1 mainly captures, anxiety, angry hostility, depression, self-consciousness, impulsiveness, and vulnerabilty, traits also referred to as emotional instability or neuroticism. PC2 mainly captures warmth, gregariousess, assertiveness, activity, excitement-seeking, positive emotions, and fantasy, traits often referred to as extraversion. PC3 mainly captures trust, straightforwardness, altruism, compliance, modesty, and tender-mindedness, traits often referred to as agreeableness. PC4 mainly captures competence, order, dutifulness, achievement striving, self-discipline and deliberation, traits often referred to as conscientiousness.PC5 mainly captures the variables fantasy, aesthetics, feelings, ideas, actions and values, traits often referred to as openness. Altogether, it seems that my component structure replicates the findings of the Big-5 / OCEAN personality traits model, which is a well-accepted model for assessing personality in modern psychology research.

```
##        PC1    PC2    PC3    PC4    PC5
## V2  -0.224 -0.090  0.156 -0.326  0.089
## V3  -0.232 -0.170 -0.119 -0.253  0.079
## V4  -0.279 -0.071  0.154 -0.230  0.119
## V5  -0.248  0.012  0.136 -0.260  0.114
## V6  -0.175 -0.231  0.090 -0.119 -0.161
## V7  -0.290 -0.008  0.166 -0.146  0.039
## V8   0.200 -0.161  0.282 -0.035 -0.171
## V9   0.135 -0.191  0.127  0.026 -0.317
## V10  0.199 -0.235 -0.163 -0.004 -0.068
## V11  0.148 -0.264 -0.077 -0.240 -0.065
## V12  0.027 -0.308 -0.069 -0.020 -0.243
## V13  0.154 -0.263  0.176 -0.072 -0.215
## V14 -0.074 -0.285  0.101  0.180  0.192
## V15  0.068 -0.187  0.179  0.046  0.456
## V16  0.023 -0.308  0.174 -0.152  0.181
## V17  0.105 -0.205  0.092  0.214  0.168
## V18  0.111 -0.206 -0.012  0.180  0.436
## V19  0.030 -0.174  0.032  0.237  0.169
## V20  0.192  0.048  0.264  0.163 -0.044
## V21  0.110  0.245  0.260 -0.099  0.096
## V22  0.206 -0.008  0.306 -0.166 -0.182
## V23  0.111  0.220  0.335  0.088  0.051
## V24 -0.034  0.201  0.300 -0.069  0.001
## V25  0.098  0.014  0.375 -0.011 -0.015
## V26  0.294 -0.010 -0.110 -0.102  0.098
## V27  0.181  0.074 -0.116 -0.358  0.043
## V28  0.244  0.128  0.001 -0.210  0.176
## V29  0.230 -0.104 -0.132 -0.297  0.152
## V30  0.289  0.049 -0.123 -0.230  0.023
## V31  0.183  0.201 -0.068 -0.131  0.224
```



```
## $loadings
##
## Loadings:
##      PC1    PC2    PC3    PC4    PC5
## V2  -0.438
## V3  -0.324        -0.245
## V4  -0.413
## V5  -0.391
## V6  -0.220 -0.240         0.128
## V7  -0.337                0.123
## V8         -0.366  0.199
## V9   0.101 -0.390
## V10  0.143 -0.201 -0.203 -0.133
## V11        -0.271 -0.167 -0.229
## V12        -0.332 -0.214
## V13        -0.415
## V14                       0.139  0.372
## V15 -0.127         0.110 -0.116  0.484
## V16 -0.240 -0.181        -0.113  0.279
## V17                              0.347
## V18         0.125                0.498
## V19                              0.316
## V20  0.163 -0.113  0.297
## V21                0.364 -0.122
## V22        -0.305  0.287 -0.109
## V23                0.420
## V24 -0.121         0.337
## V25        -0.155  0.344
## V26  0.126               -0.315
## V27                      -0.390 -0.147
## V28                0.126 -0.360
## V29               -0.104 -0.414
## V30                      -0.375
## V31         0.205        -0.306
##
##                  PC1   PC2   PC3   PC4   PC5
## SS loadings    1.000 1.000 1.000 1.000 1.000
## Proportion Var 0.033 0.033 0.033 0.033 0.033
## Cumulative Var 0.033 0.067 0.100 0.133 0.167
##
## $rotmat
##            [,1]       [,2]       [,3]        [,4]       [,5]
## [1,]  0.6231574 -0.3108074 0.25293554 -0.65698057  0.1395486
## [2,]  0.1912907  0.6458209 0.47551337 -0.06052083 -0.5626256
## [3,] -0.3089358 -0.3257671 0.83915697  0.22785910  0.2057432
## [4,]  0.6504628  0.1697974 0.03990122  0.63308154  0.3816840
## [5,] -0.2377806  0.5927553 0.06429703 -0.33466664  0.6899027
```

Question B4

As previously described I ran varimax rotation on the PCA loading matrix. This rotated matrix shows show the original variables load on the newly derived components. V1 to V7 especially load negatively on PC1 (neuroticism) with loadings of -0.438, -0.324, -0.413, -0.391, -0.220, and -0.337 respectively. This suggests that the original variables share a lot of variation and might be influenced by the same underlying construct. One can clearly see how there are 5 loading clusters on the respective principal components when looking at the varimax - rotated loading matrix.

It is possible to group the participants based on their personalities. To achieve this goal one can one clustering algorithms to organize a set of objects into groups in such a way that objects in the same group (cluster) are more similar to each other than to those in other groups.

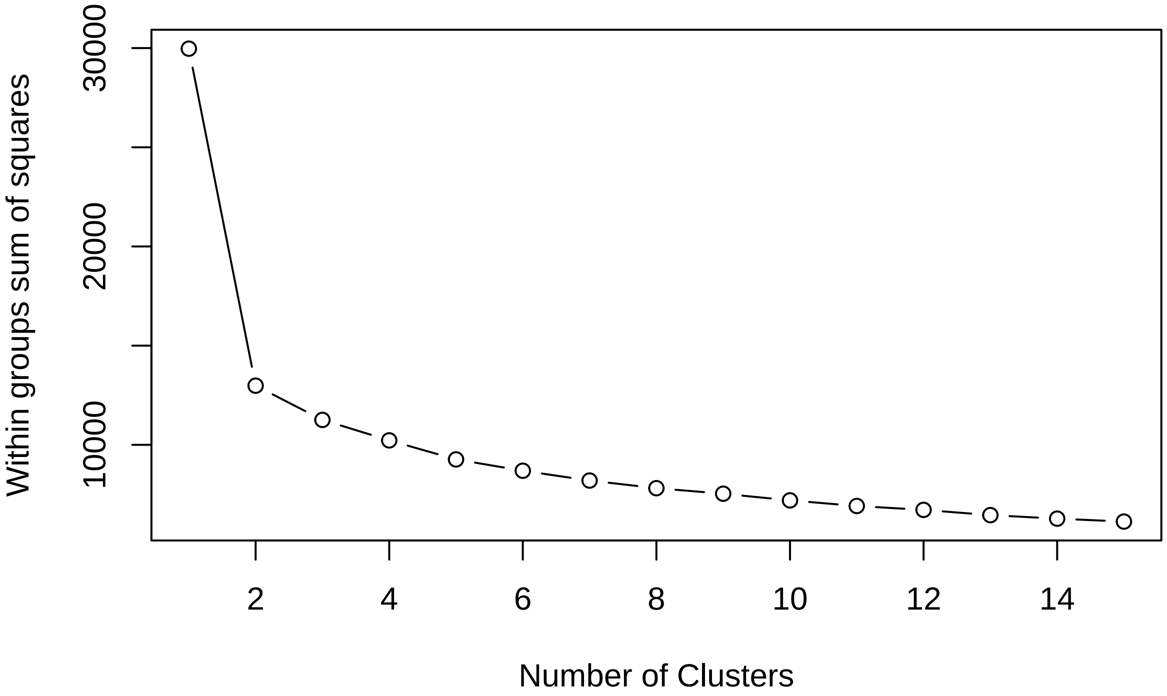
Question B5

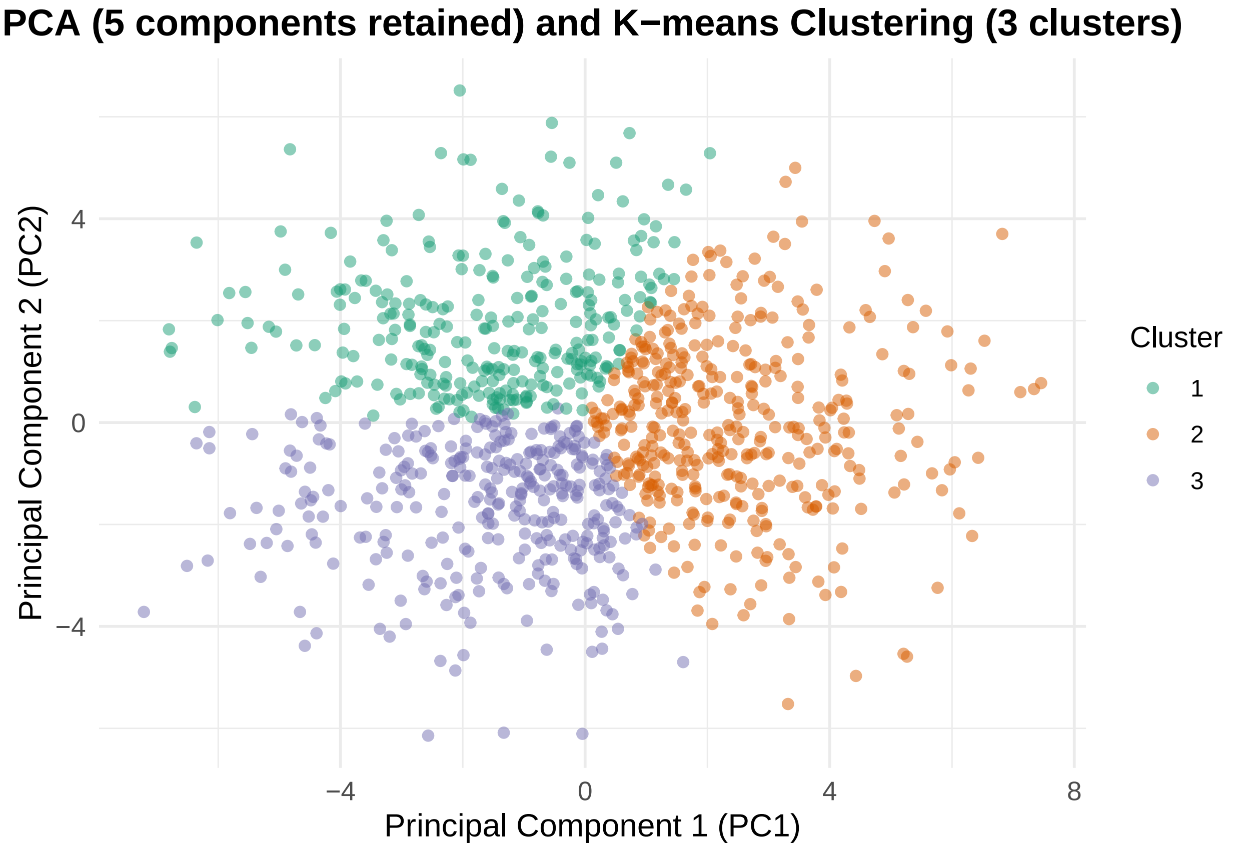
There are different clustering algorithms available to reach the goal of grouping the participants based on their personality profiles. Two prominent techniques are k-means clustering and hierarchical clustering. K-means might be efficient because it works well with a large number of observations, which is the case in our sample. It also works well with PCA-reduced data as PCA often normalizes the scale of the variables, which is beneficial since K-means uses Euclidean distance as a metric. K-means also lets us choose the number of mutually exclusive clusters. Therefore, we can look at the data and see which number of clusters makes sense by both looking at statistical properties as well as domain knowledge considerations.

Hierarchical clustering, on the other hand, creates a tree of clusters and does not require the number of clusters to be specified a priori. This technique might be suitable when our aim is to build a taxonomy of personality types. We can then see how the clusters emerged and how the taxonomy is represented by looking at the dendogram of clustering steps. Hierarchical clustering does also not require pre-specification of the number of clusters, which can be advantageous if the number of natural groupings is unknown, like in our case. However, one has to keep in mind that we have 1000 students / observations in the present case, and therefore it might be hard to extract a structure / taxonomy from the dendogram. It might even be impossile to visually plot the dendrogramm. It might therefore be reasonable to run both clustering algorithms and see if they arrive at similar results for the clustering.

Question B6

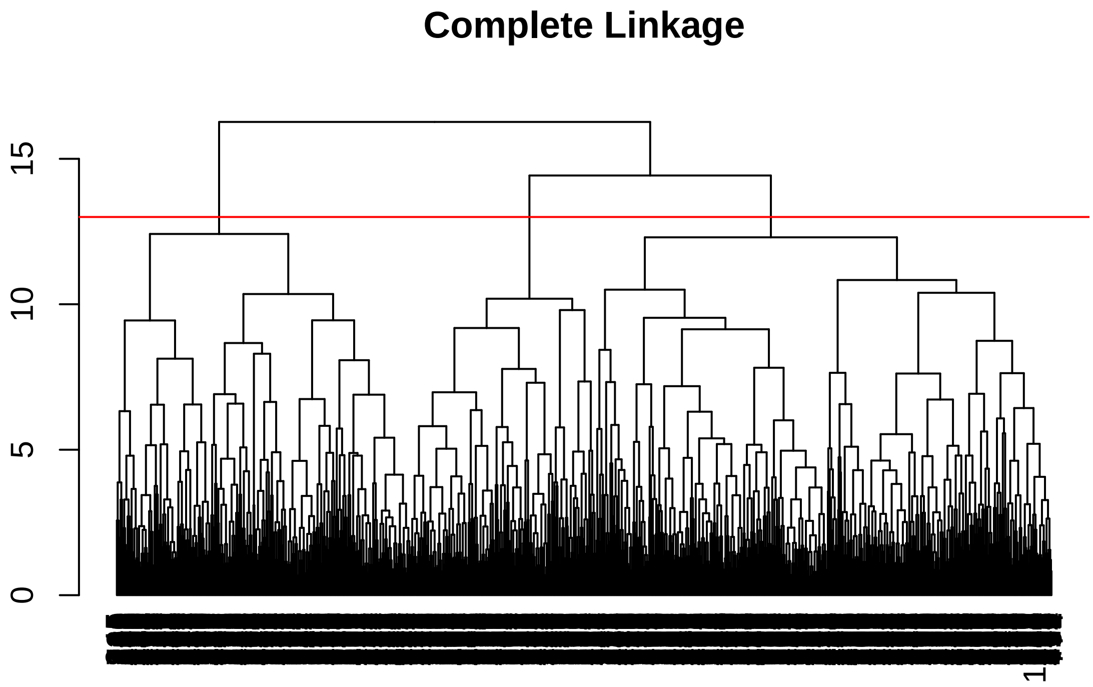
To see if there are distinct personality profiles in the data I ran the K-means clustering algorithm. The goal is to see how the new variables cluster in the reduced feature space, therefore looking at the scores of the PCA. To choose an appropriate k (number of clusters) I used the Elbow method which involves plotting the within-cluster sum of squares (WSS) against the number of clusters and picking the k at which the WSS starts to diminish at a slower rate (see plot). I also looked at the scree-plot of the clusters plotted against the first two PC's which explain most of the variance and can still be visualized in 2D space. The elbow method suggests to take 3 clusters into account. However, when looking at the scree-plot 3 clusters still seperate the profiles precisely without any overlap. When trying k = 4 there is substantial overlap between the clusters, which is why I decided to go for 3 clusters / personality profiles (see scree plot).





```
##
##   1   2   3
## 198 487 315
```

Then I can hierarchical clustering using a euclidean distance matrix between the observations in the PCA score matrix with the derived 5 PC's. The clustering tree progressed according to complete linkage. It is almost impossible to derive a precise personality taxonomy when looking at the dendogramm with 1000 observations. However, one can clearly see that there is an emerging cluster strutcure when the tree progresses. At first sight I decided to take a cut off point at 5 clusters as this seems to be a meaningful separation point to differentiate between the underlying subclusters. However, as one could also go for a cutoff point at 3 clusters, as this would still catch the variation in the personality types, and would match with the results of k-means clustering.



Question B7

When looking at cluster assignments from k-means clustering, group 1 consists of 284 subjects, group 2 of 390 subjects, and group 3 of 326 subjects. When running hierarchical clustering, group 1 consists of 198 subjects, group 2 of 487 subjects, and group 3 of 315 subjects.

Question B8

While k-means and hierarchical clustering both assigned an almost equal amount of subjects to group 3, they differ in the assignments for group 1 and 2. K-means allocated more subjects to cluster 1, whereas hierarchical clustering allocated more subjects to cluster 2. As previously described, hierarchical clustering (HC) might yield different results when cutting the dendroramm at 5 clusters, probably resulting in a more nuanced separation of cluster 3.

Question B9

To investigate the differences in personality between the clusters, I looked at the PCA scores in the newly derived coordinate system spanned by the several prinicipal components. We saw earlier by looking at the loadings of the original variables that these components capture different aspects of personality, also described as the big 5. I computed the mean for each cluster on the components / personality clusters to see if there are differences between groups (see plot). Whereas there are no differences between groups for agreeableness (PC1), conscientiousness (PC4), and openness (PC5), one can clearly see that groups differ in neuroticism (PC1), and extraversion (PC2). Group 2 shows very high neuroticism scores, whereas group 1 and 3 show rather low scores (original variables where already standardized to enable comparisons). Group 1 shows comparatively high extraversion scores, whereas group 2 shows very average scores, and group 3 is rather introverted compared to the other groups.

```
##   cluster_km neuroticism (PC1) extraversion (PC2) agreeableness (PC3)
## 1          1         -1.536421         1.95231248          0.07455748
## 2          2          2.437014        -0.02766586         -0.04027069
## 3          3         -1.576969        -1.66769037         -0.01677532
##   conscientiousness (PC4) openness (PC5)
## 1             -0.10621044    -0.03307635
## 2              0.05698235     0.07698343
## 3              0.02435782    -0.06328176
```