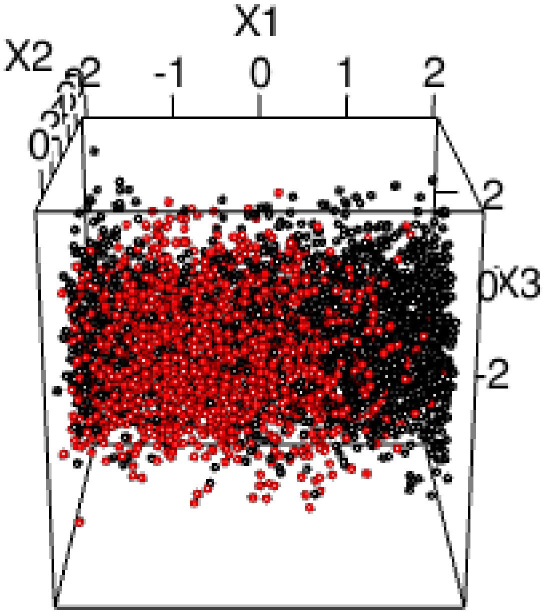
Assignment 1

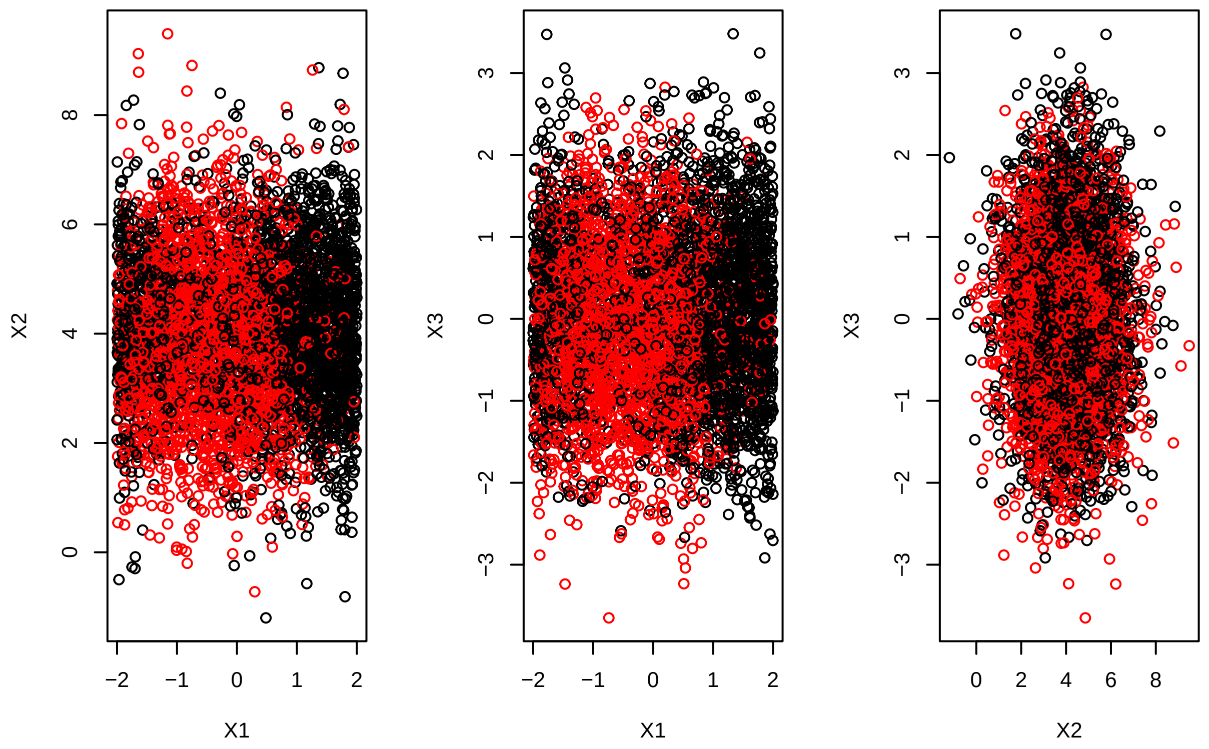
Valentin Kodderitzsch 3895157

2024-04-09

A1

From the data generating function we know that only X1 to X3 are relevant to create the binary class label Y. The following two figures illustrate the decision boundary in 3D for X1 to X3 only, selected from the training set.





We can observe a somewhat linear decision plane/boundary with class 0 towards the center and class 1 at the edge of the parameter space. There is no perfect separation between the two classes which could be explained by η (found in the data generating function) which is non-linear and has discontinuities due to the indicator function.

When discussing the expected prediction error (EPE) between K-nearest neighbors (KNN) analysis versus LASSO logistic regression (LLR), we need to take both the dimensionality of the predictors and the decision boundary into consideration. However, let's first investigate the KNN and LLR model assumptions.

KNN is a non-parametric model and thus has low bias & high variance. It makes no assumption about the decision boundary and can capture it in both non-linear and linear ways. LRR is a parametric model and is the discriminative counterpart to LDA when ignoring the LASSO penalty and therefore assumes a linear decision boundary. When considering the LASSO penalty also, then LRR is (very) high bias & low variance compared to KNN.

In the end, I expect KNN to have only marginally lower/better EPE compared to LRR, assuming 0-1 loss. Reason being that KNN can easily fit the "imperfections" of the decision plane without suffering from the curse of high dimensionality (further explained in A2). LRR is more bias/stringent about the linearity of the decision boundary. However, as we only 6 predictors, lots of observations (5000) and a somewhat linear decision plane I expect both KNN and LRR to perform similarly.

A2

In the second scenario we consider all 203 predictors. X1 to X3 explain Y, whereas X4 to X203 are just noise. It is important to know that X1 as well as all noise variables (X4-X203) are drawn from a uniform $U(-2, 2)$. Thus, any model might have difficulties discriminating between X1 and noise. X2 and X3 are drawn from a normal distribution so it should be more easily distinguishable.

In this 203 predictive variable scenario I expect KNN to perform much worse than LRR. Reason being that with 203 predictors and 5000 observations, KNN will suffer the curse of dimensionality. In other words, all observations will be far apart (using the Euclidean distance measure) in high dimensional space. Determining the class label based on the nearest "neighbor" will not work anymore as all "neighbors" are similarly far away (and none are close by) undermining the meaning of classification by proximity. As such, KNN in scenario A2 will just learn the noise and will not be able to generalize well on the test set.

On the other hand, LRR will do a good job cutting down the number of predictors from 203 to a sensible number capturing mostly the signal. If X1 will be in the predictors remains to be seen, as X1 is drawn from the same distribution as the noise. In any case, LRR will do a better job at generalizing on the test set as it is high bias compared to KNN.

A3

Consider only Y and predictors X1 to X6.

a)

Let's fit KNN with 10 fold cross validation on the training set. As previously mentioned we have 5000 observations and 6 predictive variables for a binary classification problem. Thus, all k's in the parameter grid should be odd numbers. A small k for KNN will result in low bias & high variance compared to a large k (leads to high bias & low variance). The tuning grid was constructed using the "1 to $\sqrt(n)$" heurisitc plus a couple of higher k values.

```
## k-Nearest Neighbors
##
## 5000 samples
##    6 predictor
##    2 classes: ’0’, ’1’
##
## Pre-processing: centered (6), scaled (6)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4499, 4501, 4500, 4501, 4500, 4500, ...
## Resampling results across tuning parameters:
##
##   k    Accuracy   Kappa
##     1  0.6416014  0.2772463
##     3  0.6644015  0.3231435
##     5  0.6701995  0.3356612
##     7  0.6884015  0.3729928
##     9  0.6895979  0.3755146
##    11  0.6965911  0.3899613
##    15  0.6956003  0.3885808
##    21  0.7055919  0.4094564
##    31  0.7129916  0.4253801
##    45  0.7123932  0.4256101
##    65  0.7113883  0.4248063
##   151  0.7069931  0.4172488
##   371  0.6973951  0.3966852
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 31.
```

We observe an increase-max-decrease shape for the accuracy (0-1 loss) as k increases. The optimal/max k is k = 31 as obtained by 10 fold cross validation on the training set with X1 to X6 predictors.

Now compute the EPE on the test set with k = 31. The test set can only be used once as it will otherwise be part of the model building procedure which would violate the philosophy of correct model performance estimation.

```
## [1] "KNN test set accuracy for k=31 is 0.7104 with EPE of 0.2896"
```

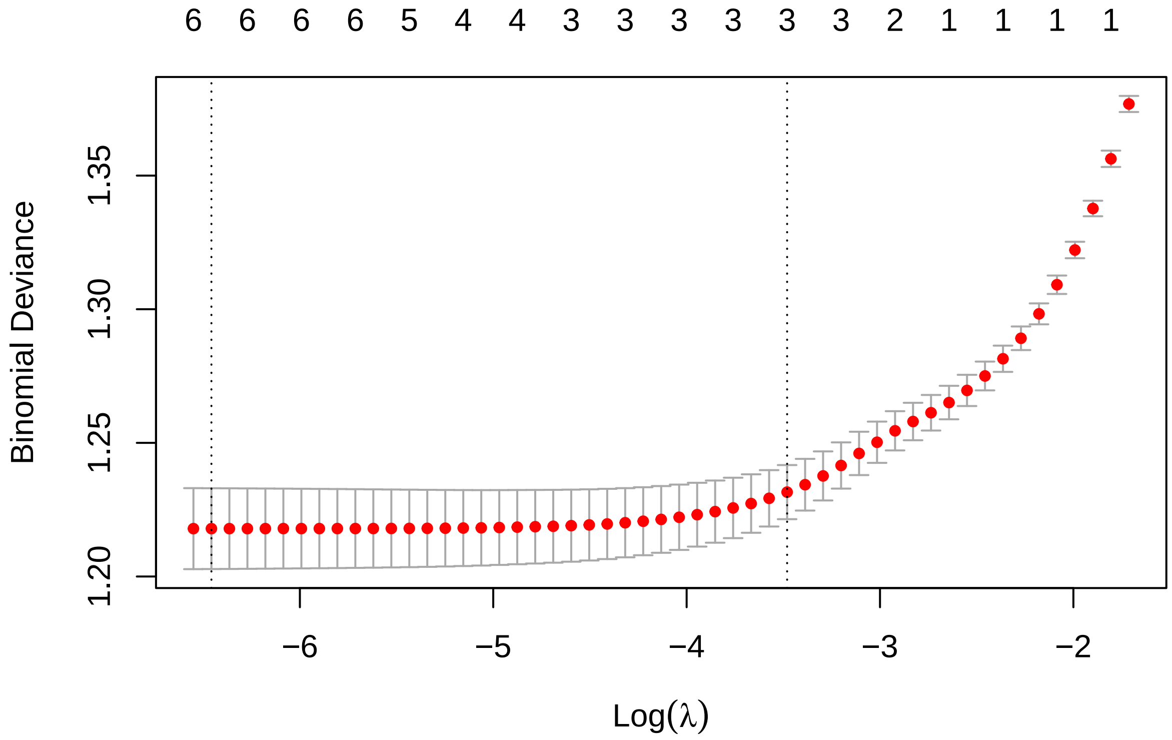
b)

Now find the optimal λ for LRR using 10 fold cross validation on the training set. To this end, use the `glmnet` package and set $\alpha = 1$ for the LASSO penalty.

```
## Loading required package: Matrix

## Loaded glmnet 4.1-8

##
## Call:  cv.glmnet(x = as.matrix(train[2:7]), y = train$Y, nfolds = 10,      alpha = 1, family = "bino
##
## Measure: Binomial Deviance
##
##       Lambda Index Measure      SE Nonzero
## min 0.001569    52   1.218 0.01512       6
## 1se 0.030797    20   1.232 0.01012       3
```



```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept) -0.005096378
## X1           0.557393534
## X2           0.050334822
## X3           0.113463627
## X4           .
## X5           .
## X6           .
```

We observe that even though the 1 standard error (1SE) λ value has a slightly higher binomial deviance (i.e. slightly worse fit), it correctly shrinks all noise variables X4 to X6 to zero.

Since we know from the set up of this exercise that only X1 to X3 are relevant, I will proceed my analysis with $\lambda = 0.030797$ chosen by the 1SE criterion instead of the minimum binomial deviance criterion.

Now compute the EPE on the test set with $\lambda = 0.030797$ (1SE).

```
## [1] "LRR (6 predictors) test set accuracy for 1SE lambda=0.0308 is 0.654 with EPE of 0.346"
```

c)

On X1 to X6 we observe the following EPEs, **KNN: 0.2896** and **LRR: 0.346**.

Compared to my initial explanation from question A1, these estimates were mostly expected. As correctly predicted in question A1, KNN has a lower EPE compared to LRR. However, contrary to my initial analysis, KNN's EPE is a good amount (almost 6% points) lower than LRR's, instead of just being "marginally" lower.

This can be explained by the non-parametric nature of KNN which makes the model low bias & high variance compared to LRR. I believe that I have over-emphasized the "linearity" of the decision boundary when analyzing it in question A1. As such, I believed that both LRR and KNN should share a fairly similar, "linear" decision boundary. However, in the low dimensional scenario (X1 to X6 predictors) KNN was able to pick up the non-linearities of the decision boundary leading to a noticeably lower EPE compared to LRR.

Interestingly enough, LRR was able to correctly penalize all noise variables X4 to X6. So I under-emphasized LASSO's abilities to filter out the noise correctly. As such, LRR with the 1SE λ is very high bias & low variance compared to KNN, as it always picks up the relevant predictors X1 to X3, whilst assuming a linear decision boundary.

A4

Consider Y and all predictors X1 to X203.

a)

Follow the same procedure as described in A3a). Only difference is that the data set has all 203 predictors now.

```
## k-Nearest Neighbors
##
## 5000 samples
##  203 predictor
##    2 classes: '0', '1'
##
## Pre-processing: centered (203), scaled (203)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4499, 4499, 4500, 4499, 4501, 4500, ...
## Resampling results across tuning parameters:
##
##   k    Accuracy   Kappa
##     1  0.5260061  0.04150301
##     3  0.5383945  0.06578719
##     5  0.5436077  0.07328595
##     7  0.5431977  0.06959165
##     9  0.5493957  0.08121324
##    11  0.5479909  0.07735185
##    15  0.5448021  0.06818180
##    21  0.5563921  0.08735069
##    31  0.5673862  0.10565825
##    45  0.5651906  0.09528778
##    65  0.5756022  0.11285278
##   151  0.5830074  0.11880286
##   371  0.5620026  0.05946994
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 151.
```

Similar observation as A3a). We observe an increase-max-decrease shape for the accuracy (0-1 loss) as k increases. The optimal/max k is k = 151 as obtained by 10 fold cross validation on the training set with 203 predictors.

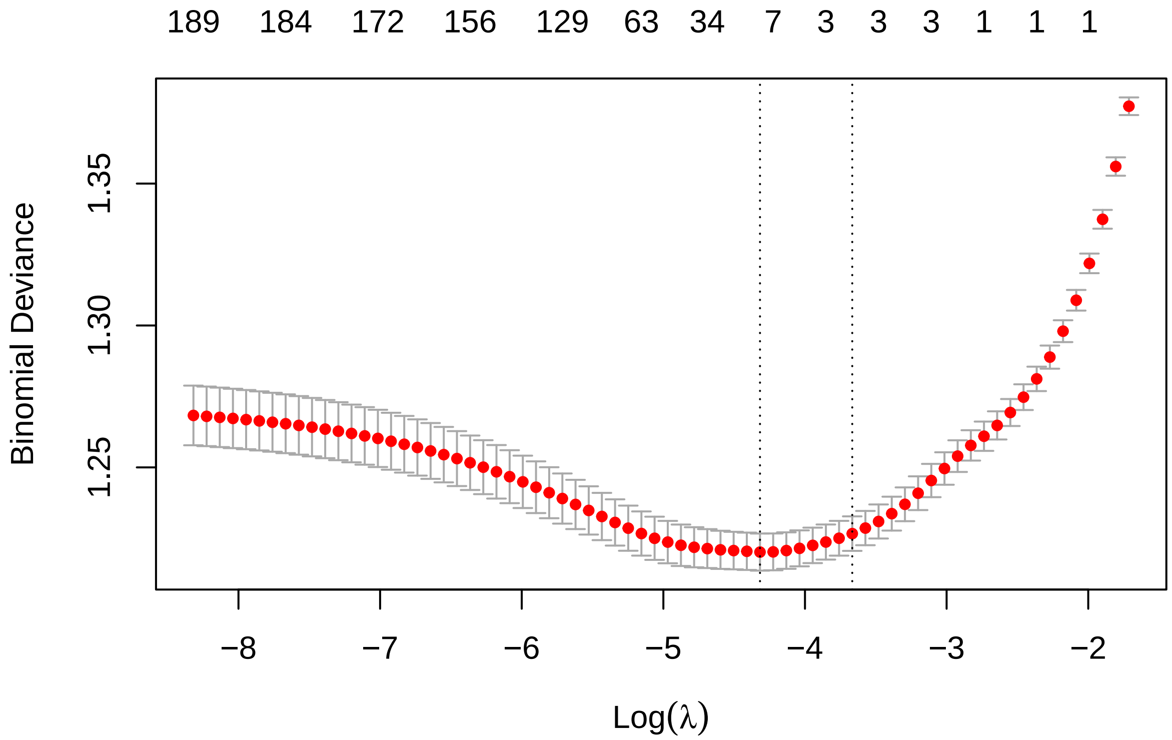
Now compute the EPE with k = 151 on the test set.

```
## [1] "KNN test set accuracy for k=151 is 0.5688 with EPE of 0.4312"
```

Note that KNN with k=65 (instead of k=151) will likely perform similarly on the test data and it would be less computationally expensive. But I wil move forward with k=151 as this was selected by the model building procedure.

b)

```
##
## Call:  cv.glmnet(x = as.matrix(train[-1]), y = train$Y, nfolds = 10,      alpha = 1, family = "binomi
##
## Measure: Binomial Deviance
##
##      Lambda Index Measure       SE Nonzero
## min 0.01333    29   1.220 0.006536       8
## 1se 0.02557    22   1.227 0.006080       3
```



Now compute the EPE on the test set with $\lambda = 0.0255684$ (1SE).

```
## [1] "LRR (203 predictors) test set accuracy for 1SE lambda=0.0256 is 0.6576 with EPE of 0.3424"
```

c)

On X1 to X203 we observe the following EPEs, **KNN: 0.4312** and **LRR: 0.3424**.

Compared to my explanation for question A2, these answers were expected. KNN performed significantly (almost 9% points) worse compared to LRR.

As previously explained in A2, KNN suffered from the curse of dimensionality. With 203 predictors classification by proximity will not work anymore, as all "neighbors" are similarly far away. Thus, proximity starts to become meaningless and KNN will not be able to properly distinguish between the signal and noise. This explains the EPE of 0.4312, which is the same as saying that KNN has a missclassification rate of 2 in 5. So not that good model performance.

LRR on the other hand, was able to correctly shrink all noise predictors to zero when using the 1SE criterion for λ on the test set. I did expect LRR to significantly cut down the noise in the predictors. But LRR exceeded my expectation in how well it did so as it correctly identified relevant predictors only. This is reflected in LRR's EPE of 0.3424 which is significantly better compared to KNN. This was correctly explained in A2.

Now let's compare KNN and LRR individually for the case with few predictors versus many predictors.

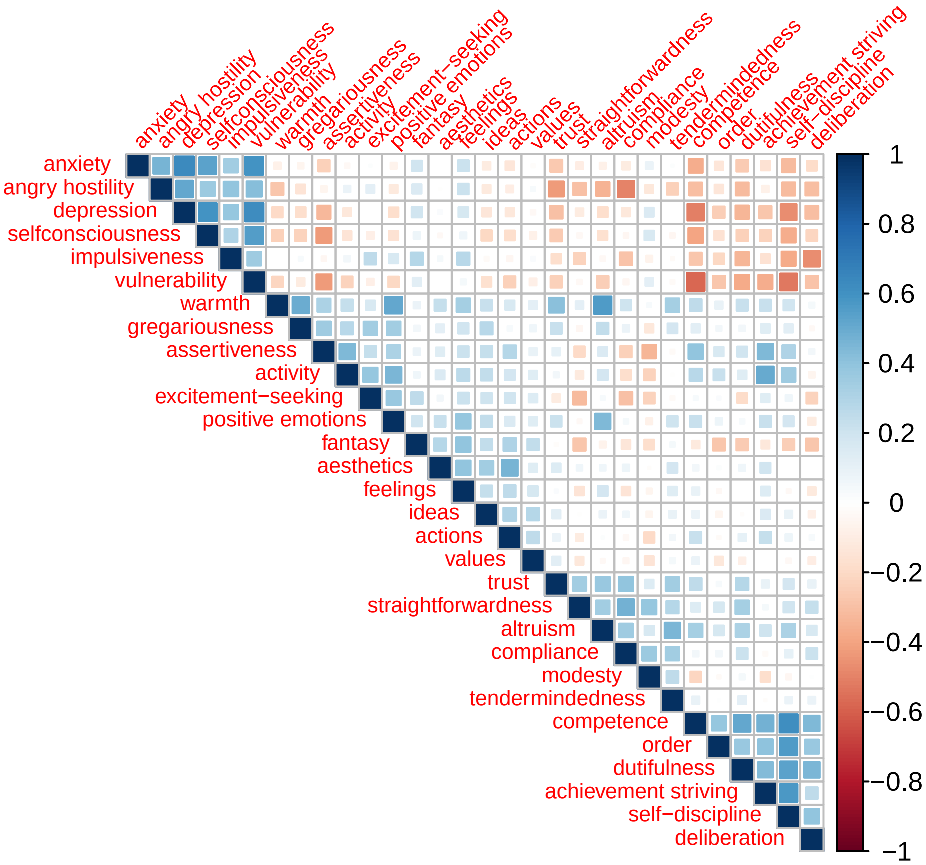
KNN in the 6 predictor case has an EPE of 0.2896, and then an EPE of 0.431 in the 203 predictor case. As previously explained this was expected as KNN is a non-parametric, low bias & high variance model, that suffers from the curse of dimensionality. Thus, KNN generalizes well in the case of few explanatory variables but has poor model performance when the number of predictors is high.

LRR has an EPE of 0.346 in the 6 predictor case and an EPE of 0.3424 in the 203 predictor case. I expected LRR to perform similarly in both cases. However, I did not expect LRR to have such a closely matching EPE in both cases. This could be explained by the 1SE criterion for λ which correctly identifies the relevant predictors in both cases. As such both LRR models are nearly identical in both cases resulting in closely matching EPE values. This further shows that LRR with the 1SE criterion for λ is a high bias & low variance model that performs well on the given dataset.

B0 Load data

Load the data. We observe that all variables have already been standardized, as mentioned in the assignment.

Plot the correlation.



As mentioned in the assignment many variables are highly correlated.

B1

Q: Do you think that reducing the variables to a smaller set of (new/derived) variables may be a good idea? Which statistical technique can be used to achieve such a dimension reduction? Explain why this technique is appropriate.

Reducing the variables to a smaller set of potentially new variables may be a good idea because:

- 30 variables are more difficult to interpret and keep an overview of compared to 5 or 6 variables

- Some of the 30 variables are highly correlated, e.g. anxiety, depression, vulnerability, so there is some redundancy
- Compressing redundant variables into new variables might lead to a small loss of information but it will help identify the "big picture personality traits", ie underlying structure in the data, rather than getting lost in the details

Principle component analysis (PCA) can be used to achieve such a dimension reduction.

PCA is appropriate because:

- It summarized the variance of the data set using a smaller number of variables/components
- All new variables are uncorrelated
- The top PCs can potentially capture the majority of the variance, leading to minimal information loss whilst reducing the dimensionality of the data set

PCA works by creating new components which are linear combinations of the original variables. Said linear combinations can be difficult to interpret if the loadings are not rotated. But for the sole purpose of reducing the dimensionality of the personalities data set, PCA is an appropriate technique.

B2

Q: How many new/derived variables should be computed to capture the most important part of the information in the original variables? Use at least four different methods to select the number of new/derived variables. Explain each method and thoroughly justify your final decision (e.g., by providing relevant figures).

To summarise, I will use the top 5 principle components (PCs), i.e. 5 new variables instead of 30, to capture the most important information of the given dataset. I will justify my decision making in detail at the end of B2.

To identify the top PCs I used the following rules:

1. Kaiser's rule (a simple rule for an upper bound)
2. Cattell's scree plot (a simple rule for a lower bound)
3. Horn's parallel analysis (to account for sampling fluctuations - a more complex/sophisticated rule)
4. MAP (to account for a correlated dataset - a more complex/sophisticated rule)

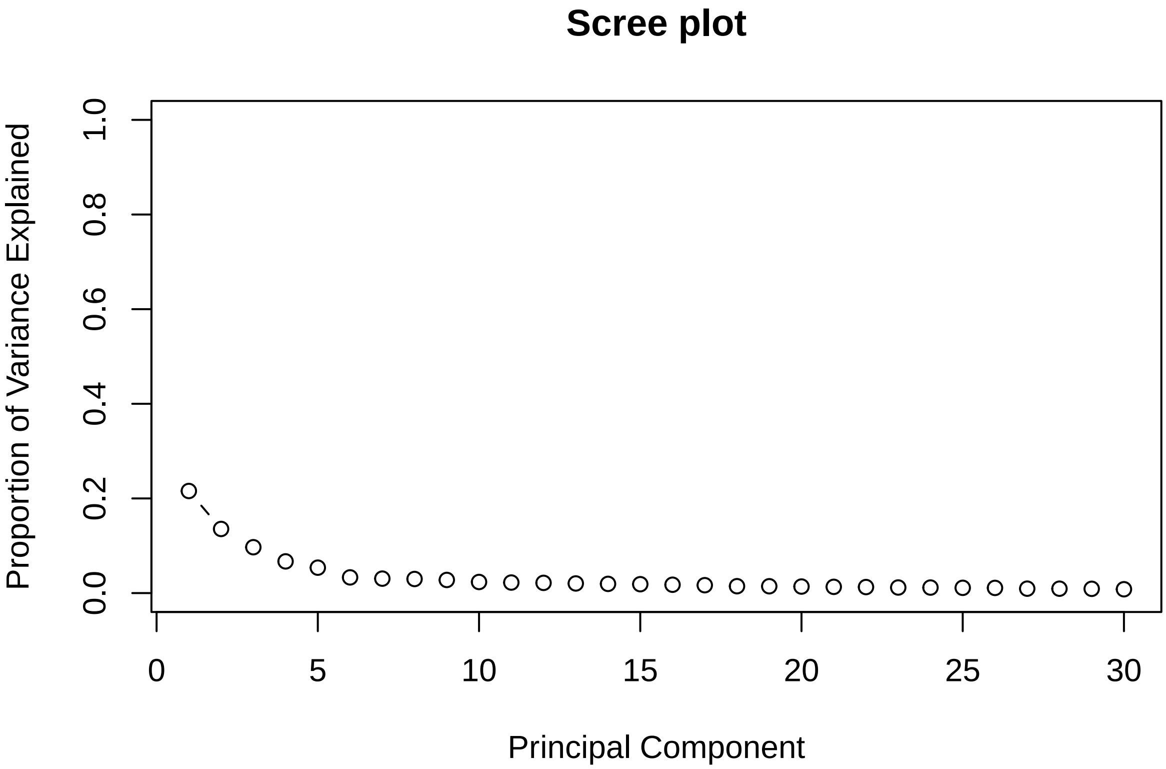
Kaiser's rule

Choose the PC's with eigenvalues greater than 1. This gives an upper bound, as Kaiser's rule tends to identify more "top" PCs compared to other rules.

```
## [1] "Kaiser's rule returns the top  5  PCs"
```

Scree plot

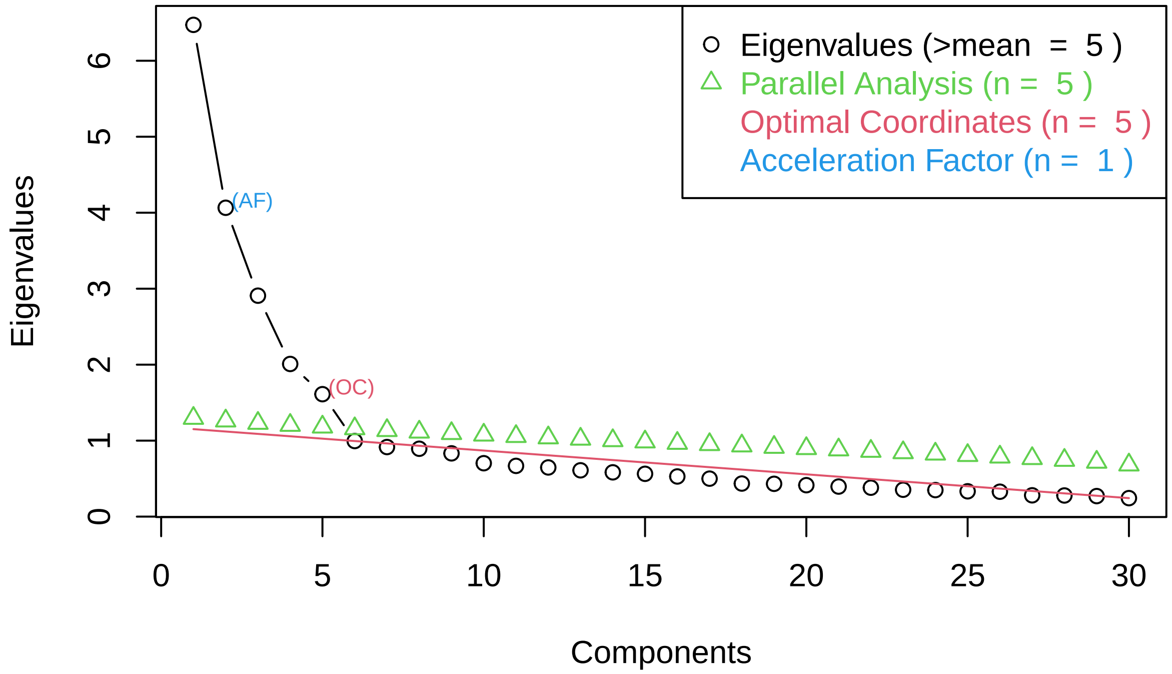
The scree plot is a more subjective rule that tends to give a lower bound, so fewer "top" PCs compared to other rules. In the below scree plot the proportion of variance explained (PVE) is plotted against each PC. As the first PC captures the most variance, there is a downwards trend and we are looking for the point of diminishing returns, i.e. the "elbow".



Based on the "elbow" of the scree plot, maybe choose the top 5 or 6 principle components.

Horns

Horn's parallel analysis is based on a simulations. In the first step a random dataset is generated that follows the same distribution as the original personality traits dataset. Then PCA is performed on the simulated data and the eigenvalue of PC is calculated. In a final step, the mean eigenvalue over all simulations per PC are compared to eigenvalues from the PCA on the original personality traits dataset. Said mean eigenvalues represent a cutoff value, similar to $\lambda > 1$ in Kaiser's rule.



For Horn's we want to stay above the green "parallel" line, representing the cutoff value. So in our case that would mean choosing the top 5 PCs.

MAP

Velicer's MAP stands for minimum average partial test. In the first step, the partial correlations are computed. Then PCA is performed on the original dataset. In a third step, the average squared partial correlation is computed per PC, where n is the nth component. Then, the optimal number of PCs is chosen by identifying the nth PC with the lowest average squared partial correlation.

MAP also suggests top 5 PCs. The code output was intentionally ommited here to save space.

Discussion

All 4 methods agreed on the same answer (top 5 PCs), despite having different approaches and levels of complexity. This convergence underlines the robustness of the decision making process, as different methods usually return different numbers of top PCs. Therefore, I am confident that using the top 5 PCs for further analysis sufficiently captures the core variability of the original dataset.

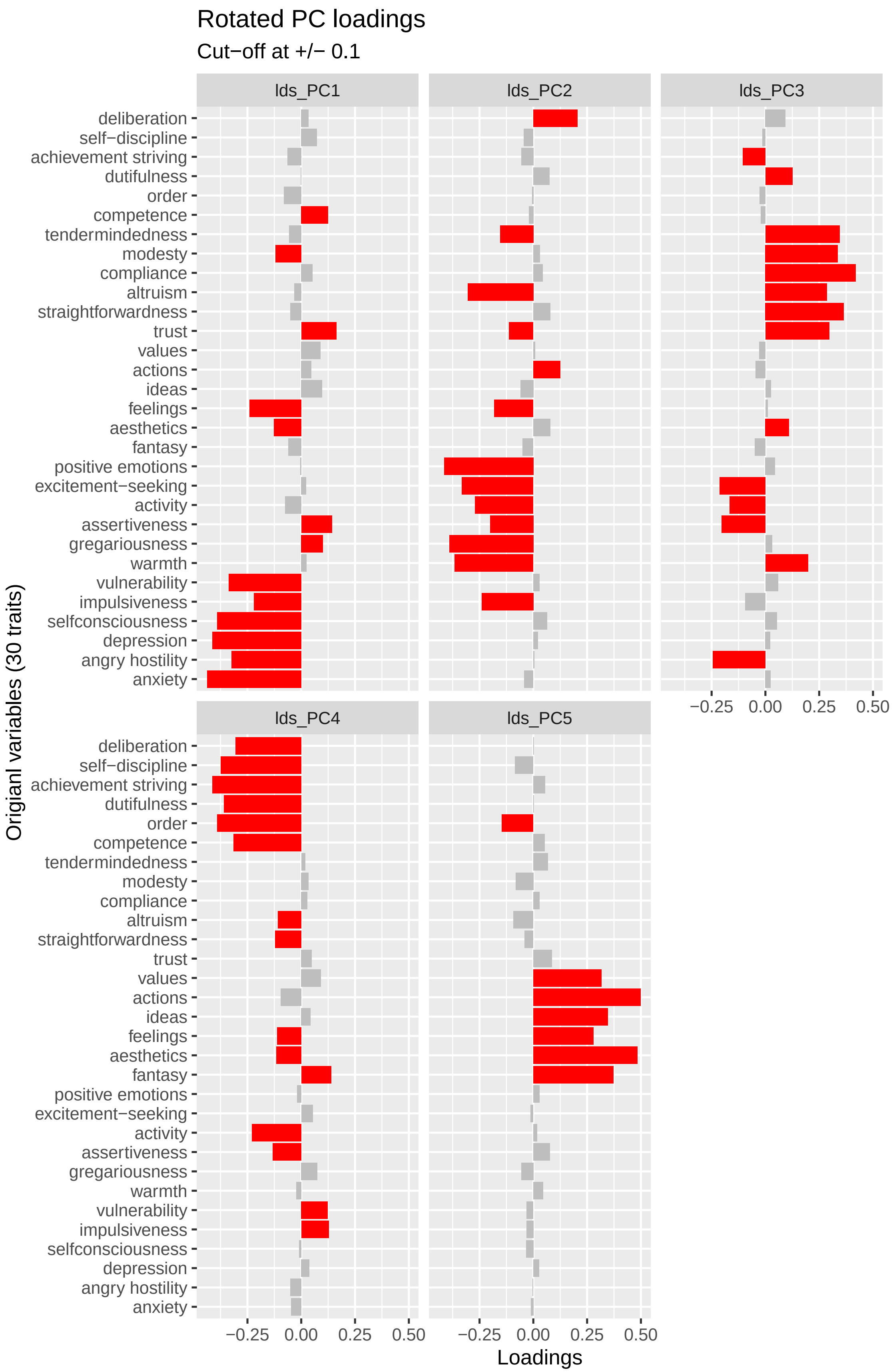
B3

Q: Can you give a meaning to these new/derived variables?

Yes, it is possible to give meaning to these new variables, i.e the top 5 PCs. But for that we need to perform a rotation such that most loadings have a small weight and only a few have a high weight.

To do so, I performed an orthogonal rotation using varimax. I also considered an oblique rotation but the resulted loadings were less interpretable compared to orthogonal rotation. Therefore, I decided to move forward with an orhtogonal rotation. I also considered quartimax for the orthogonal rotation but here the results were not that interpretable either. Therefore, I decided to use orthogonal rotation using varimax in the end.

When plotting the loadings I left the cutoff at the default value of the varimax function which is +/- 0.1.


Rotated PC loadings
Cut−off at +/− 0.1
lds_PC1
lds_PC2
lds_PC3
lds_PC4
lds_PC5
deliberation
self−discipline
achievement striving
dutifulness
order
competence
tendermindedness
modesty
compliance
altruism
straightforwardness
trust
values
actions
ideas
feelings
aesthetics
fantasy
positive emotions
excitement−seeking
activity
assertiveness
gregariousness
warmth
vulnerability
impulsiveness
selfconsciousness
depression
angry hostility
anxiety
−0.25
0.00
0.25
0.50
Loadings
Origianl variables (30 traits)

The rotated loadings as shown in the above figure allign well with the big five personality traits. Based on our data the PC loadings (i.e. weights for the original variables) can be interpreted in the following way.

- Loadings of PC1: **Neuroticism** (with high weights/loadings variables being anxiety, depression, self-consciousness, vulnerability)
- Loadings of PC2: **Extraversion** (with high weights/loadings variables being gregariousness, positive emotions, warmth, excitement seeking)
- Loadings of PC3: **Agreeableness** (with high weights/loadings variables being compliance, tender mindedness, straightforwardness, modest)
- Loadings of PC4: **Conscientiousness** (with high weights/loadings variables being achievement striving, order, self discipline, dutifulness)
- Loadings of PC5: **Openness** to experience (with high weights/loadings variables being action, aesthetics, fantasy, ideas)

B4

Q: Can you somehow quantify the degree to which the new/derived variables capture the information in the original variables?

Yes, use the proportion of variance explained (PVE) by the top 5 PCs before rotation.

```
## [1] "The top 5 PCs (no rotation) explain  56.8863414970269 % of the total variance of the dataset"
```

The PVE is different for the top 5 PCs after rotation is performed.

```
##
## Loadings:
##                      PC1    PC2    PC3    PC4    PC5
## anxiety              -0.438
## angry hostility      -0.324        -0.245
## depression           -0.413
## selfconsciousness    -0.391
## impulsiveness        -0.220 -0.240         0.128
## vulnerability        -0.337                0.123
## warmth                      -0.366  0.199
## gregariousness        0.101 -0.390
## assertiveness         0.143 -0.201 -0.203 -0.133
## activity                    -0.271 -0.167 -0.229
## excitement-seeking          -0.332 -0.214
## positive emotions           -0.415
## fantasy                                    0.139  0.372
## aesthetics           -0.127         0.110 -0.116  0.484
## feelings             -0.240 -0.181        -0.113  0.279
## ideas                                             0.347
## actions                      0.125                0.498
## values                                            0.316
## trust                 0.163 -0.113  0.297
## straightforwardness                 0.364 -0.122
## altruism                    -0.305  0.287 -0.109
## compliance                          0.420
## modesty              -0.121         0.337
## tendermindedness            -0.155  0.344
## competence            0.126                -0.315
## order                                      -0.390 -0.147
## dutifulness                         0.126 -0.360
## achievement striving               -0.104 -0.414
## self-discipline                            -0.375
## deliberation                 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
```

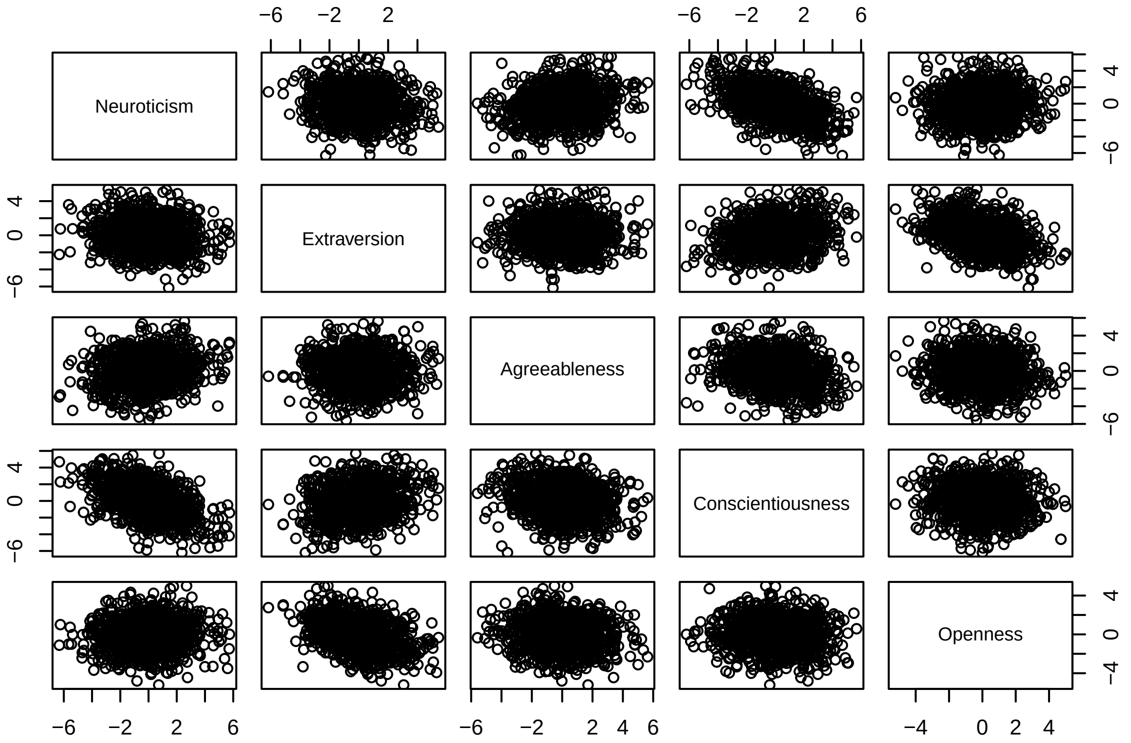
```
## [1] "The top 5 PCs explain 16.7 % of the total variance of the dataset"
```

This drop in PVE is expected as it is a consequence of the rotation process. However, it will not negatively affect the rest of our analysis as the new rotated components capture are more specific aspect (i.e. focus on the big 5 personalities) of the data compared to the unrotated components.

Q: Is it possible to group the study participants in terms of their personalities? To this end, use the new/derived variables (and not the original ones).

Yes, it is possible. To do so we need to perform matrix multiplication with the top 5 loadings, to project the original variables onto the space spanned by the top 5 PCs. Then, we can perform different grouping, ie. clustering techniques. I will further explain this in question B5.

When plotting the dataset post projection using the 5 new variables (big five personality traits) we observe that they form a single cloud of points. This was somewhat expected as the new PCs are uncorrelated. However, this might have an implication on the number of clusters which I will discuss later on.



B5

Q: Which statistical technique(s) can be used to group the participants? Explain why it is/they are appropriate.

For this assignment I interpret "grouping participants" as partitioning participants into distinct, homogeneous subgroups (clusters) based on similar personality traits.

To this end we can use: 1) K means clustering (centroid based) 2) Hierarchical clustering (connectivity based) 3) Gaussian mixture clustering (distribution based). We will use said clustering algorithms on the newly derived variables (rotated PCs). This will allow us to describe the clusters/subgroups formed by the 1000 participants (from the original dataset) in terms of the big 5 personalities.

K means might be an appropriate algorithm as it a linear algorithm and thus fast to run on any dataset. As it is a rather simple method it will definitely not overfit the data. However, as it assumes that all clusters are of the same size and non-overlapping, there is a chance of underfitting, i.e. not correctly capturing the underlying clusters. Plus, we need to make an assumption about the number of clusters k, meaning we need to perform some hyper parameter tuning.

Hierarchical clustering might be a good alternative as we do not need to make an assumption about the number of clusters k. It is also a very flexible algorithm and the resulting dendogram can easily be interpreted. However, the algorithm is slow to run and very sensitive to the chosen distance metric and linkage type. So, some experimentation for the different distance metrics and linkage types is required as well as exploring a top-down vs a bottom-up approach.

Gaussian mixture clustering (GMC) could also be an appropriate algorithm, as it can be seen as the generalized case of K means clustering. Or in other words, K means can be seen as a special case of the Gaussian mixture clustering with “high bias, low variance” as K means assumes spherical, uncorrelated Gaussian clusters. As such, GMC is a very flexible method. However, there is a risk of overfitting.

It is hard to tell which clustering technique might be the most fitting to the data so I will try out all of them. However, the fact that the new variables form a single cloud might be an indicator that we have few rather than many clusters.

B6

Q: How many groups are there? How did you determine this? Thoroughly justify your answer (e.g., figures). If you identified more than one technique in the previous question, choose two and compare the techniques and the obtained results.

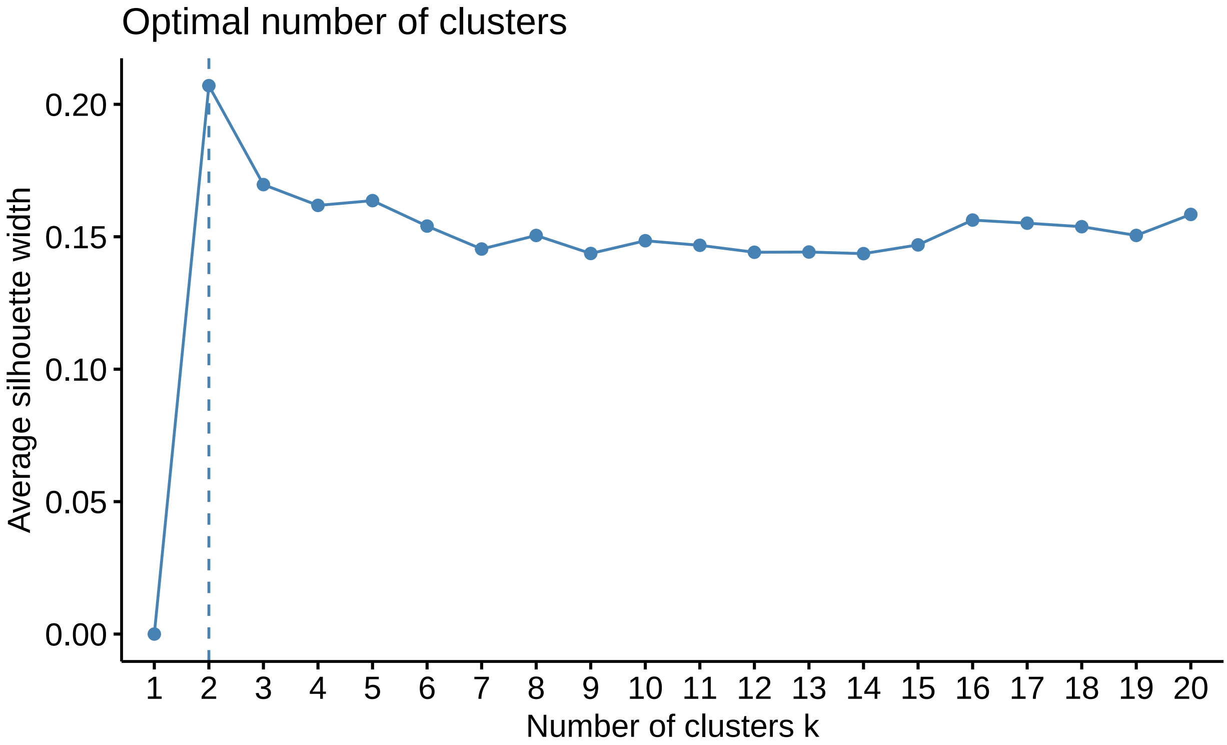
Based on my analysis there are 2 subgroups in the dataset of 1000 participants. I chose K means and GMC as my two main techniques. Hierarchical clustering was my sensitivity analysis to help me confirm the number of clusters k identified by my main methods. Another reason why I chose hierarchical clustering as my secondary method is because of question B9.

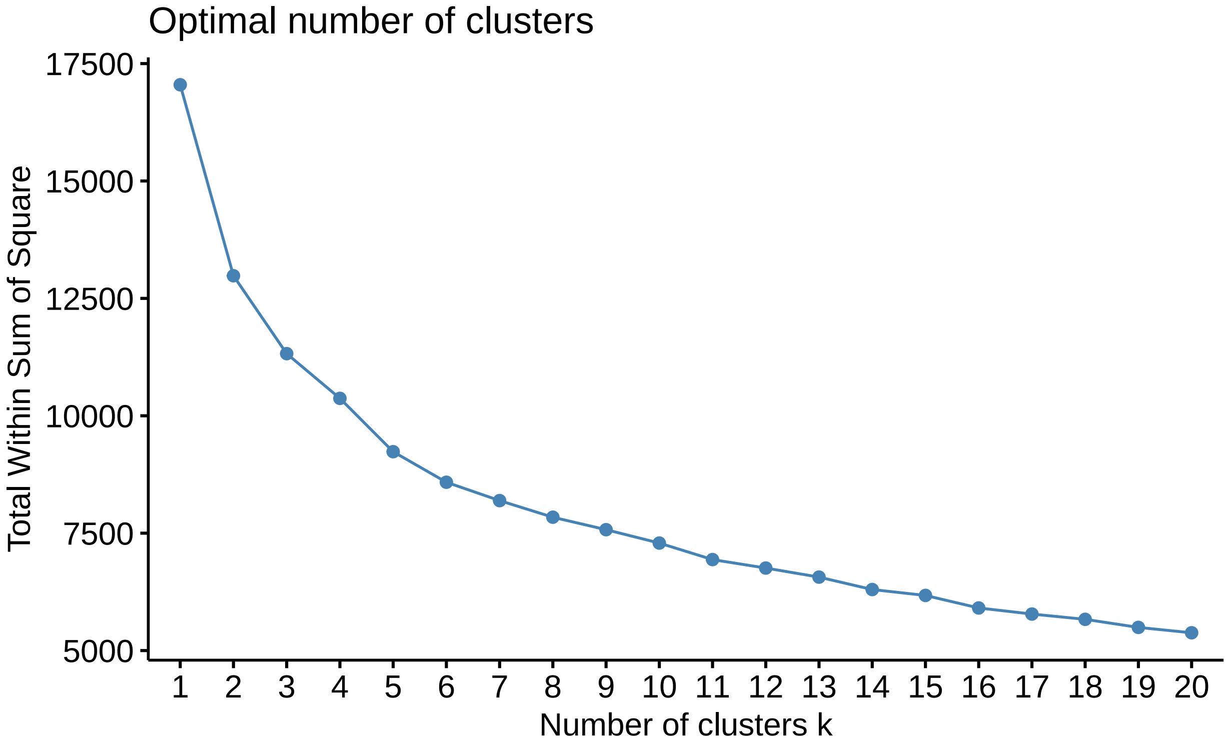
To determine the optimal number of clusters for K means I used two methods: 1) Scree plot of the sum of squared errors over k clusters and 2) a Silhouette plot. For GMC, I plotted the BIC of all models (differentiated by their covariance matrices) over k clusters to determine the optimal number of clusters.

For hierachical clustering I visually inspected the dendograms and fitted the following linkage types: Complete, single, average and Ward. As the distance measures for hierarchical clustering are often context depended I just left it at the default (Euclidean distance measure), since I do not have access to subject specific expert knowledge. Moreover, all variables are mean centered, standardized and uncorrelated so the Euclidean distance measured should work without any issues.

I will compare K means against GMC in the discussion section of B6 as well as in B8.

K means

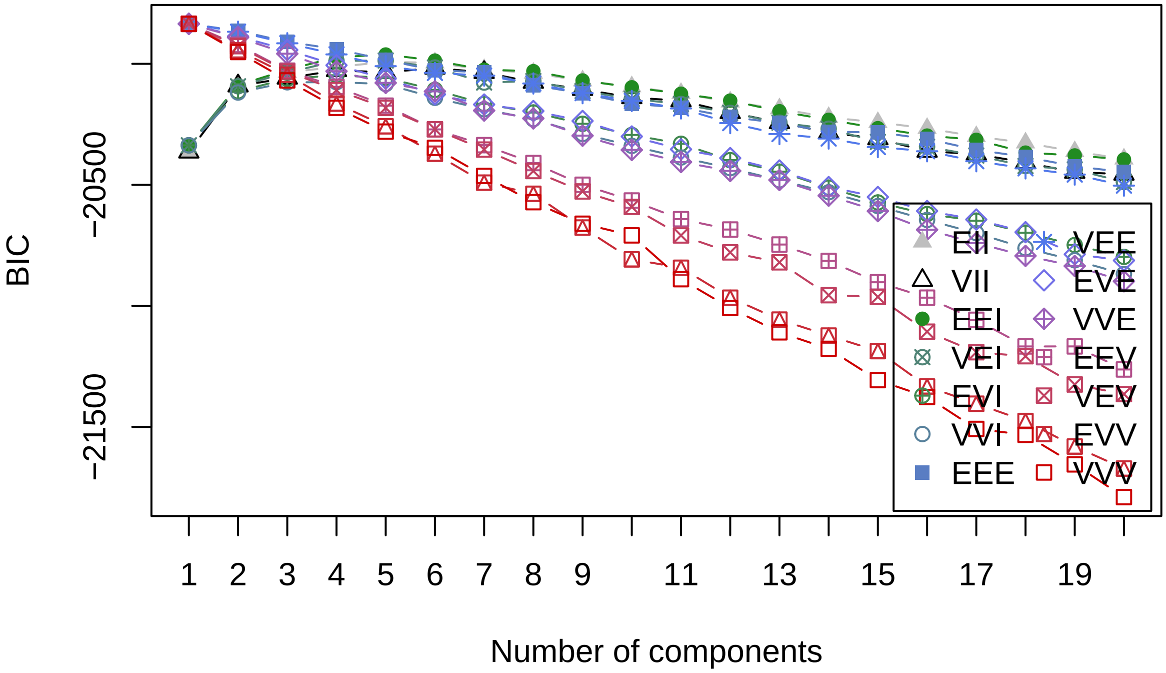




Scree plot suggests an "elbow" at k = 5. Silhouette plot indicates the optimal number of clusters to be k = 2.

Gaussian Mixture Clustering

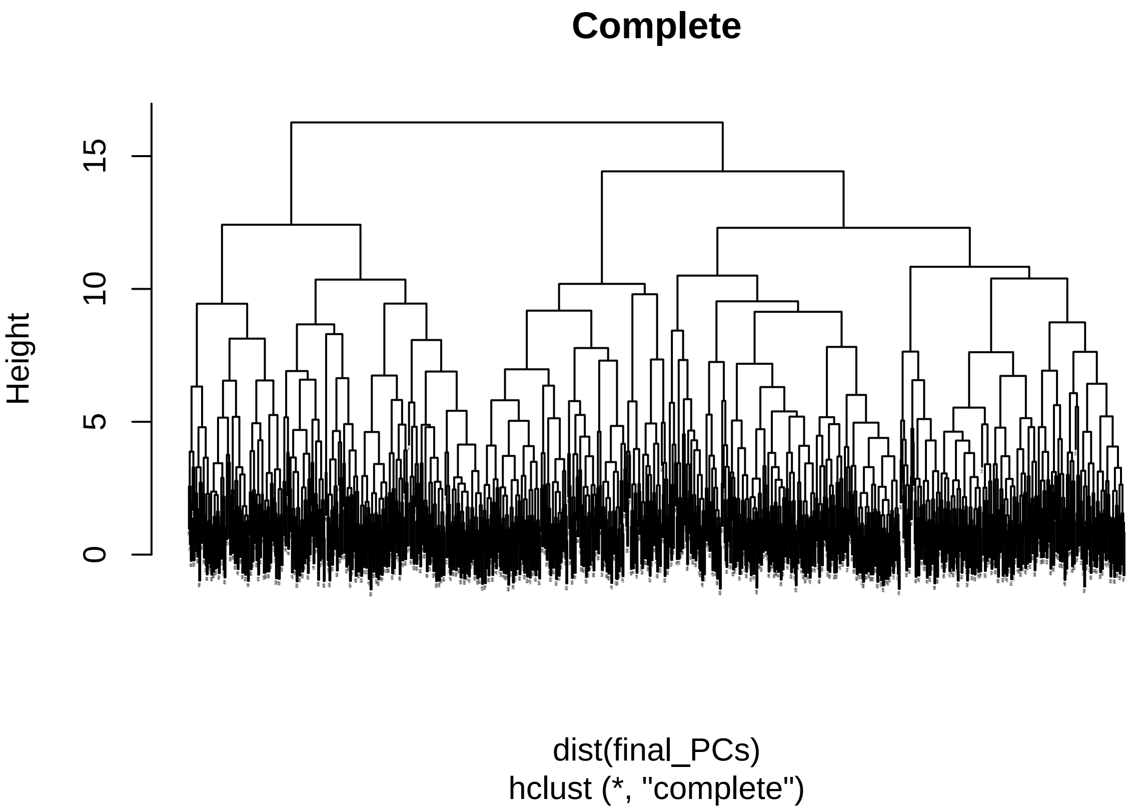
```
## Package 'mclust' version 6.1
## Type 'citation("mclust")' for citing this R package in publications.
```



```
## [1] "Optimal covariance type and number of clusters are:  EEE,1"
```

EEE means ellipsoidal distribution and equal volume, shape and orientation. An ellipsoid cluster means correlated clusters, however, the GMC algorithm only identified one cluster as its optimum. Nonetheless, the second best BIC value is for k=2 and model EEE, so choosing said model might be an interesting alternative as its BIC value is still very close to the optimal BIC value.

Hierachical clustering



All dendograms indicate either 4 or 2 clusters except for the single linkage method. Having 5 clusters is unlikely as the heights are too close together. Having 3 clusters is also a possibility, e.g. for the complete linkage or both of the Ward linkages. I chose to display the complete linkage as comparing clusters via their maximal inter cluster dissimilarity seemed appropriate. This is analogous to distinguishing cliques in a high school cafeteria by their clique leader, which results in more pronounced group differences.

Discussion

Let's compare the results obtained by K means versus GMC. K means suggests either 2 or 5 clusters, whereas GMC suggests 1 or 2 clusters. As previously mentioned, K means is special case of GMC, with spherical, uncorrelated Gaussian clusters. As such K means is more biased as it makes stricter assumptions about the underlying clusters (ie shape and how many). However, both K means and GMC seem to agree at k=2 clusters.

The dendograms suggest either 4 or 2 clusters. Therefore, I decided to identify k=2 clusters as all three methods agree on said value.

B7

Q: How large is each group?

Different methods suggest different group sizes, but K means and GMC result in an approximate 1:1 split between the two groups.

K means

```
## [1] "The 2 groups are of size: "
```

```
## [1] 497 503
```

There were 1000 participants in total and using K means both group are pretty much the same size.

Gaussian Mixture Clustering

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EEE (ellipsoidal, equal volume, shape and orientation) model with 2
## components:
##
##  log-likelihood    n df      BIC       ICL
##       -9841.998 1000 26 -19863.6 -20322.16
##
## Clustering table:
##   1   2
## 431 569
```

Using GMC one group is a bit larger than the other but the 2 groups are still roughly the same size. This is expected as the clusters are now ellipsoids instead of perfect spheres like in K means.

Hierachical clustering

Hierachical clustering using the complete linkage suggest more unbalanced classes. However, when changing the linkage method to Ward, the classes become roughly equal sized. This is because Ward linkage approximates clusters obtained by K means.

Dendroid with complete linkage gives an unbalanced class distribution.

Discussion

Based on my main analysis method (K means and GMC) I conclude that both subgroups are roughly of the same size.

B8

Q: If you identified more than one technique, what are the main differences between your chosen methods?

In this section I will compare K means against GMC. As previously mentioned K means is a special case of GMC. Before discussing their differences I want to briefly cover their similarities.

Both K means and GMC are iterative algorithms with the number of clusters k being a pre-determined hyper parameter. Both algorithms start by randomly assigning all data points into k subgroups/centroids. Then the cluster centroids are computed and all data points are again assigned to the a cluster based on their proximity to the closest centroid.

The main difference between GMC and K means is that GMC makes no implicit assumption about the covariance matrix Σ. Instead it tries out multiple covariance matrices, resulting in different cluster shapes, sizes and orientations. Therefore, GMC is low bias, high variance compared to K means and there is a potential risk to overfit the data. K means on the other assumes independence resulting in uncorrelated, spherical Gaussian clusters. Therefore, K means is high bias, low variance compared to GMC.

B9

Q: What are the main differences between the groups in terms of personality?

As mentioned in B6 I will use K means and GMC to characterize the 2 subgroups instead of using hierarchical clustering.

K means

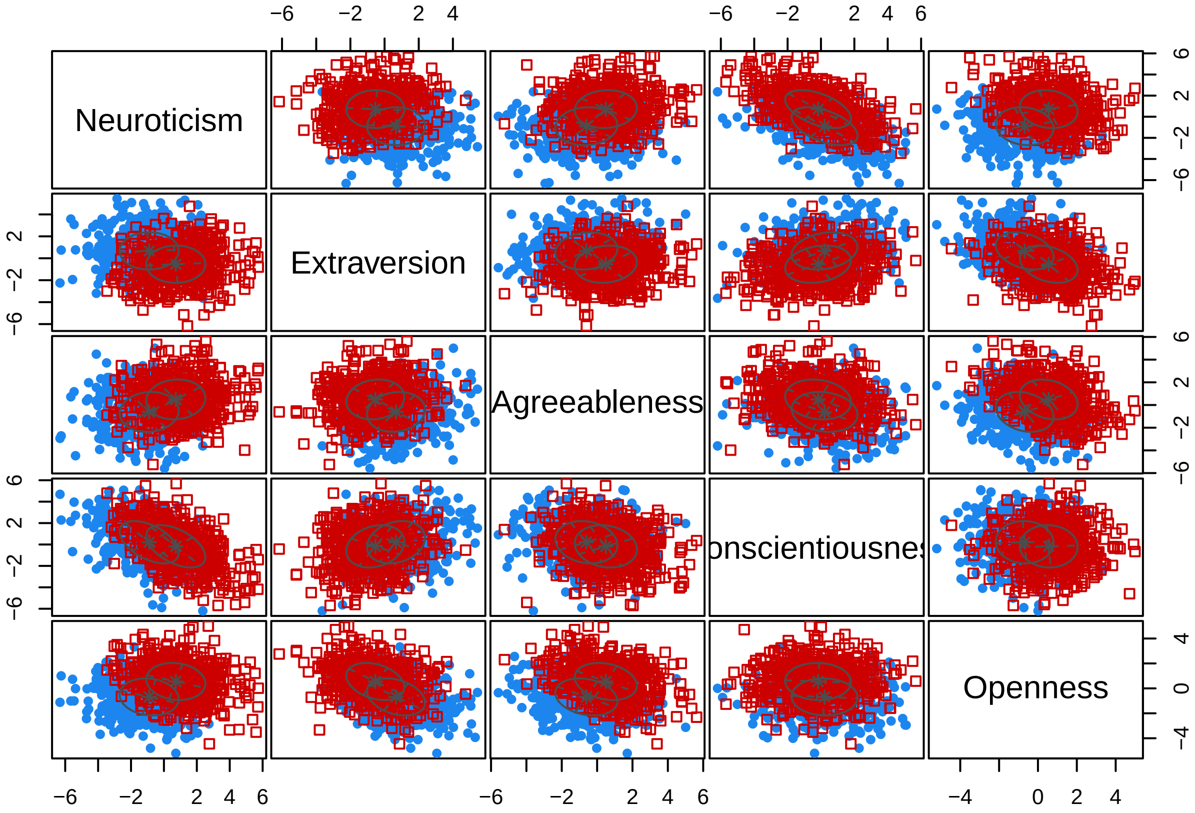
```
##   Neuroticism Extraversion Agreeableness Conscientiousness   Openness
## 1   -1.328230    0.4799913    -0.5380290          1.331778 -0.2422343
## 2    1.312386   -0.4742657     0.5316112         -1.315892  0.2393448
```

Overall, the biggest difference between the 2 groups (based on the centroid values) are that one group is well composed and conscientious whereas the other group is anxious and careless/not conscientious.

Let's describe the first group in more detail. The first group is not neurotic (i.e well composed), extroverted, not agreeable (i.e. critical), conscientious/organized and not open/cautious to new experiences.

The second group is neurotic/anxious, introverted, agreeable, not conscientious (i.e careless) and open to new experiences.

Gaussian Mixutre Clustering



Based on the clusters plotted by the GMC we can observe similar differences in the 2 subgroups as described by K means. However, the plot also shows that the centroids are quite close together, indicating that the 2 subgroups might not be that distinctly different from each other after all.

The biggest difference as shown by the GMC plot is the Neuroticism variable. It seems like there is a clear distinction between one group feeling more anxious whereas the other feels more composed. However, when considering the other variables the differences are not that big as shown by the overlapping cluster clouds.

Discussion

In conclusion, we can say that the differences in terms of personalities are not that distinct between the 2 groups. However, there is enough evidence to suggest that the main difference between the 2 groups is the Neuroticism variable, i.e. one group is more anxious whilst the other is more composed. As a secondary difference the Conscientiousness variable can also be considered.