Assignment 1

Group 11

The Date

1. Problem Statement: PCA

The file drugsrecovery.txt provides data on recovery status of patients after administration of different doses of two different drugs, L and R. The recovery status is measured as a percentage drop in body pathogens pre- and post-drug administration. A larger percentage drop implies better recovery. The administering of the drugs, at each of the dose levels, is assumed to not interfere with recovery levels for previous and/or subsequent dose(s). 100 participants took part in the study. Variables L500 to R4000, respectively refer to drug L at a dose level of 500 micrograms to drug R at 4000 micrograms. The ID is a patient's hospital identification number. Perform a principal components analysis to: I. Determine the appropriate number of components that can be used to effectively summarize the information in the data. Explain how you settled on the reported number of components. II. If possible, provide an interpretation for the chosen sample principal components III. Comment on the (bi-)plot for the first two components

2. Descriptive Statistics

check missing values and impute (not needed here), take away ID column, check if all columns are int

```r
drugs <-read.delim("data/drugsrecovery.txt", header = TRUE, sep="",dec = ".")
sub_drugs <- subset(drugs, select = -c(ID))
sub_drugs <- data.frame(sub_drugs)
str(sub_drugs)
```

```
## 'data.frame':    100 obs. of  8 variables:
##  $ L500 : int  15 10 10 10 10 20 15 5 15 10 ...
##  $ L1000: int  20 15 15 15 10 20 15 5 15 10 ...
##  $ L2000: int  25 5 30 5 5 20 15 5 15 5 ...
##  $ L4000: int  30 15 30 5 25 5 35 10 55 35 ...
##  $ R500 : int  15 15 15 5 15 15 20 5 15 5 ...
##  $ R1000: int  20 20 15 10 5 20 20 10 15 10 ...
##  $ R2000: int  20 20 25 5 5 15 20 15 5 5 ...
##  $ R4000: int  30 30 40 25 65 35 25 20 25 30 ...
```

2. Assumptions

? scaling? princomp vs prcomp?

The function princomp() uses the spectral decomposition approach.

The functions prcomp() and PCA()[FactoMineR] use the singular value decomposition (SVD).

According to R help, SVD has slightly better numerical accuracy. Therefore, prcomp() is the preferred function.

3. Method

PCA explained?

```
prin_comp <- prcomp(sub_drugs, scale = TRUE)
summary(prin_comp)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     1.9822 1.2721 0.9876 0.68321 0.58317 0.56204
## Proportion of Variance 0.4911 0.2023 0.1219 0.05835 0.04251 0.03949
## Cumulative Proportion  0.4911 0.6934 0.8153 0.87368 0.91619 0.95568
##                            PC7     PC8
## Standard deviation     0.44734 0.39303
## Proportion of Variance 0.02501 0.01931
## Cumulative Proportion  0.98069 1.00000
```

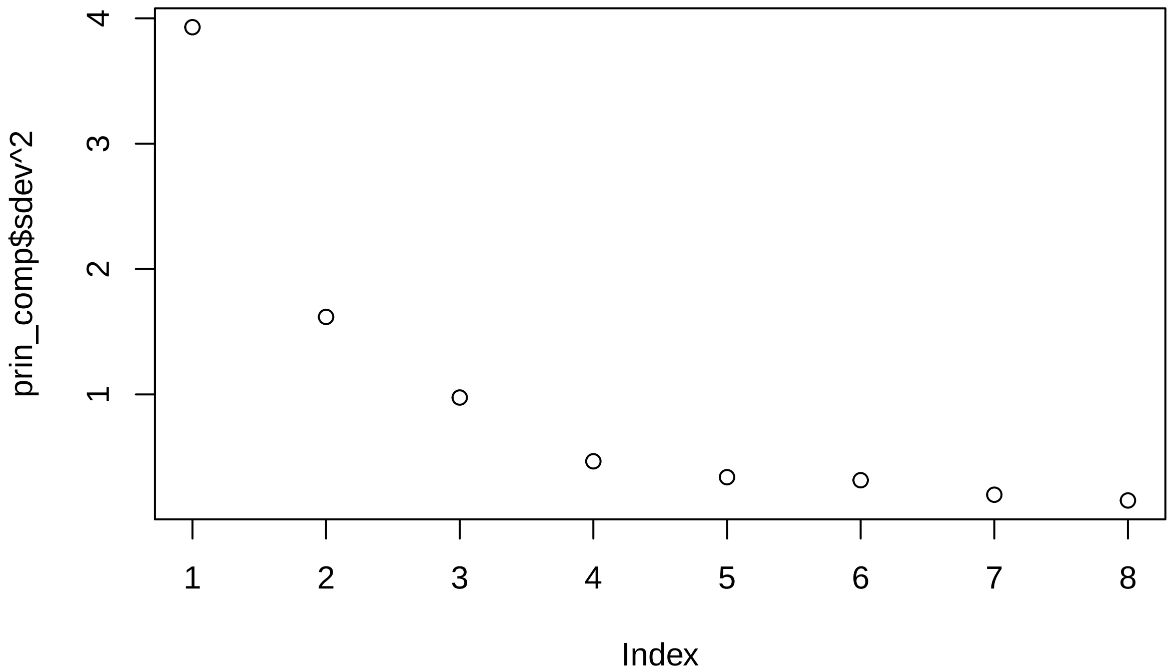
- extract values based on different approaches: extract P C 0 s to explain a given percentage of the variance • scree plot: plot the eigenvalues in decreasing order and find the elbow that distinguishes the mountain from the debris • retain only P C 0 s with eigenvalue larger than one (only for standar- dized data) • Horn's Parallel procedure: compute eigenvalues associated with many simulated uncorrelated normal variables - retain the ith PC if the corresponding eigenvalue is larger than the 95th percentile of the distribution of the ith largest eigenvalue of the random data (same idea as the previous rule but taking random variation into account)

```
library(factoextra)
```

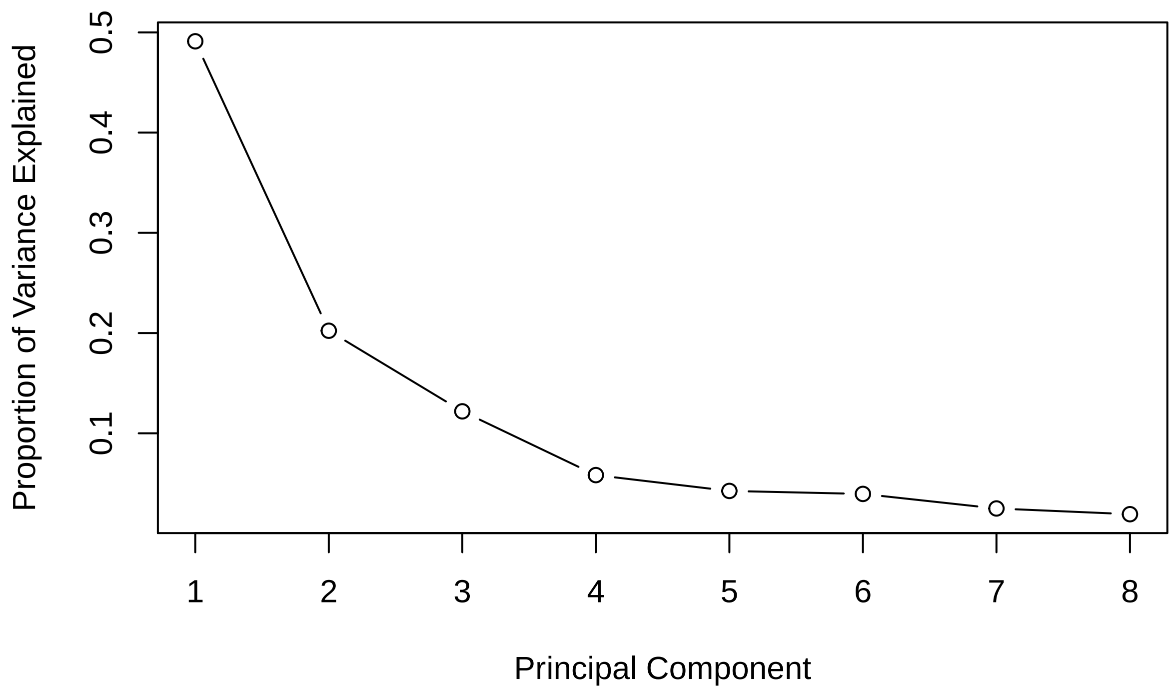
```
## Loading required package: ggplot2
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

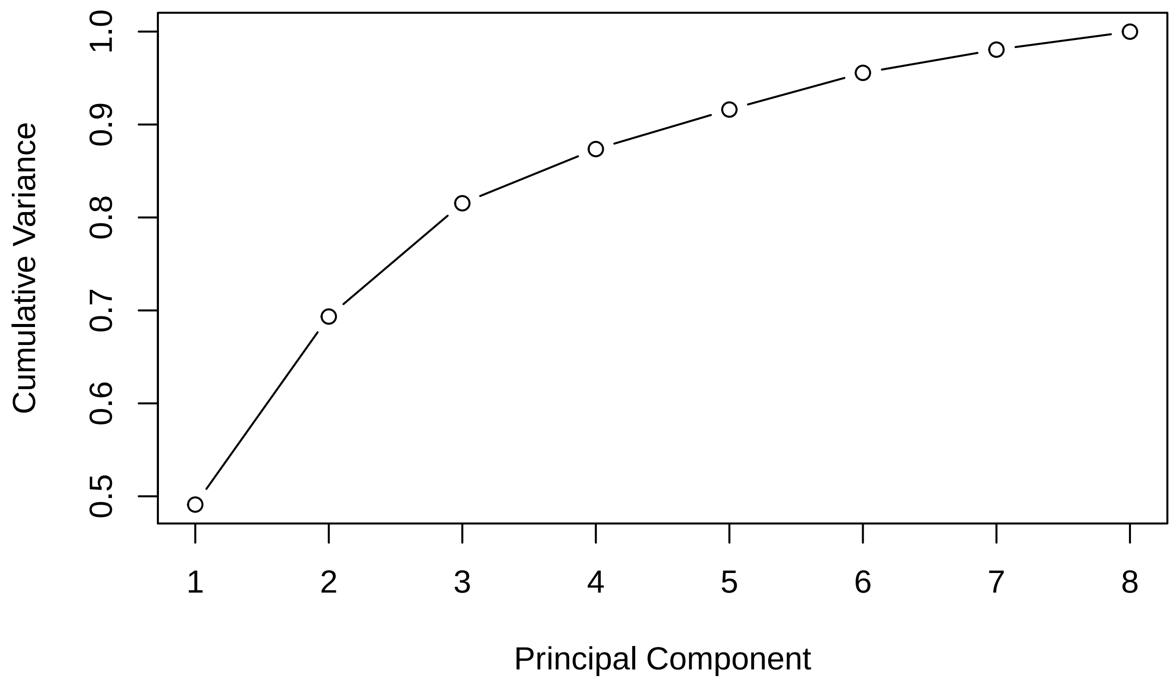
```
explained_var <-round(((prin_comp$sdev)^2)/sum((prin_comp$sdev)^2),4)
plot(prin_comp$sdev^2)
```




```r
plot(explained_var, xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     type = "b")
```




```r
plot(cumsum(explained_var), xlab = "Principal Component", ylab = "Cumulative Variance", type="b")
```




```r
res <- get_pca_ind(prin_comp)
res$contrib
```

```
##           Dim.1        Dim.2        Dim.3        Dim.4       Dim.5
## 1    3.546555e-01 2.868730e-01 5.444047e-01 4.679115e-03 0.029134066
## 2    2.215425e-02 7.259821e-01 8.414383e-02 2.915215e+00 2.587112086
## 3    1.373238e-01 1.203532e-01 1.224019e+00 3.426503e-01 0.347408867
## 4    1.156908e+00 7.024421e-01 6.748547e-04 1.022988e+00 2.849743085
## 5    5.258379e-01 4.439042e-02 3.016082e+00 7.370451e-02 0.146501728
## 6    1.909055e-01 1.743659e+00 2.820865e-01 4.055945e-02 1.750166859
## 7    1.687660e-01 6.599537e-01 7.107504e-03 1.335354e+00 0.247832170
## 8    1.800589e+00 2.272785e-02 4.841151e-01 6.245866e-01 0.541592034
## 9    3.137885e-04 2.298107e-01 1.249968e+00 8.774599e-01 1.661916198
## 10   9.331224e-01 2.173712e-03 6.170645e-01 7.161089e-01 0.265815755
## 11   4.722623e-01 1.085155e+00 1.740443e-01 1.762709e+00 1.030263703
## 12   1.441095e-01 1.779808e-03 2.806958e-01 3.669936e+00 0.259583641
## 13   8.088786e-01 1.722829e-01 4.570859e-01 4.461175e+00 0.176973740
## 14   2.382923e+00 3.572667e-02 3.259693e+00 1.235478e+00 0.243153385
## 15   9.455214e-02 3.048715e-01 1.596095e-01 1.069842e-01 1.530208702
## 16   1.360909e-01 7.748150e-03 2.705147e-02 3.485643e+00 1.040221792
## 17   3.997110e-01 7.438180e-04 1.264865e+00 1.904930e-01 3.804242375
## 18   2.056593e+00 1.270255e-01 4.389836e+00 4.040712e+00 0.127064056
## 19   3.563917e-01 2.596659e-01 7.066775e-01 1.090319e-01 0.558578568
## 20   1.705286e+00 1.085314e-02 2.457894e-01 5.426226e-02 0.018588666
## 21   3.824708e-01 4.645839e-02 1.431376e+00 4.280949e-01 0.505749503
## 22   1.377012e+00 9.195899e-01 2.879987e-03 3.888330e-01 0.887739882
## 23   1.445635e+00 2.480752e-01 1.529163e+00 1.662401e-02 0.569184286
## 24   2.297662e-01 4.895801e-01 8.961276e-01 2.105779e-01 0.334988104
## 25   3.665001e-03 2.026708e-01 7.154978e-02 2.216916e-02 0.316628955
## 26   2.022203e-01 3.250967e-02 7.021323e-02 1.426976e+00 0.471415645
## 27   5.606884e-01 3.748564e-01 1.114774e+00 2.932430e-01 1.387479117
## 28   2.710437e-01 3.577516e+00 2.200245e+00 3.214670e+00 3.180269365
```

```
## 29  2.960099e-01 1.502618e+00 1.212543e-01 8.107064e-01 0.726365346
## 30  1.126919e-01 2.598782e-01 7.049087e-01 2.047190e+00 0.218942308
## 31  4.129731e-02 2.153102e-01 3.189341e+00 7.643400e+00 0.806695486
## 32  3.722343e-02 9.955815e-03 8.356995e-01 6.422793e-01 0.023204964
## 33  3.959041e-02 4.638254e-01 6.198420e-01 2.394160e+00 1.231663415
## 34  6.287586e-01 3.652492e+00 1.044242e+00 3.064720e-01 0.058117784
## 35  3.277504e+00 4.403021e+00 2.926945e+00 2.013908e+00 5.233336558
## 36  4.514643e-04 8.692577e-01 7.957214e-02 1.705850e+00 1.049220203
## 37  2.937501e-01 2.476024e+00 2.166420e+00 5.353448e-02 0.590299298
## 38  2.125361e-01 1.553575e-01 6.051004e-02 2.032391e+00 0.508150741
## 39  2.680325e+00 3.298811e-02 2.650777e-03 9.761621e-02 0.008567574
## 40  6.721653e+00 6.443428e+00 6.908858e-02 3.271095e+00 0.009475762
## 41  1.751690e+00 2.177940e-01 1.408235e-02 4.079374e-03 0.250553633
## 42  1.163816e+00 1.407541e-01 3.000955e+00 6.456446e-01 1.513889234
## 43  8.774578e-03 6.359646e-01 3.659935e+00 4.591337e-02 0.164479485
## 44  1.012921e+00 4.215378e-01 2.654711e+00 1.046572e-01 2.704650722
## 45  9.514612e-02 5.159522e-01 4.720299e+00 1.460909e-01 1.777704549
## 46  1.214181e+00 7.422425e-01 7.929630e-03 2.491964e-01 1.412361941
## 47  1.498035e+00 2.060486e+00 1.612438e-01 4.895989e-02 0.085648534
## 48  1.047408e+00 1.427472e+00 3.490121e-01 7.227403e-02 2.416023585
## 49  5.283901e-01 1.736039e-04 1.070063e+00 1.409903e-01 0.611121366
## 50  1.707531e-01 3.367715e+00 2.445068e-01 5.119606e-01 0.253270443
## 51  2.401362e+00 3.236364e-03 6.557694e-02 1.548018e-02 0.101799302
## 52  1.969959e+00 9.200141e-03 1.299306e+00 3.707281e-01 0.093545335
## 53  9.427460e-01 1.642061e+00 5.391921e-01 5.746415e-01 0.437541954
## 54  1.405857e-01 3.103899e+00 2.343256e-02 7.167285e-01 3.307804787
## 55  7.670828e+00 3.309942e+00 2.990117e+00 1.231760e-03 0.555584842
## 56  2.611395e+00 7.060268e-02 2.297719e-01 5.996924e-02 0.449311455
## 57  4.233600e-01 7.984626e-02 1.892946e+00 1.254347e-03 0.976564133
## 58  4.196844e-01 4.381563e-02 2.737450e-01 1.849122e-04 0.080324470
## 59  3.428139e-01 9.921649e-01 1.546160e+00 4.475436e-06 0.620544380
## 60  2.591525e+00 1.050693e-01 5.085072e+00 1.283683e-01 2.040596068
## 61  9.549959e-02 1.044498e+00 7.707756e-02 7.643109e-01 1.695735725
## 62  3.732393e-01 8.324915e-03 2.558272e-02 4.549659e-01 0.543244863
## 63  1.298986e+00 4.221012e-01 5.212561e-02 1.660245e+00 0.031345560
## 64  1.400378e+00 9.072896e-02 4.129191e-03 8.080997e-01 0.267886498
## 65  6.953943e-01 1.381990e+00 2.504226e-01 2.097721e-01 0.062682795
## 66  3.562236e-03 5.571107e+00 1.856522e+00 5.867668e-01 0.461124027
## 67  5.190524e+00 2.624606e+00 4.171255e+00 4.619024e-01 1.773642297
## 68  1.315464e+00 2.278028e-08 1.055585e+00 9.346362e-02 0.967316157
## 69  2.788505e-01 1.304998e-01 4.800931e-02 1.073969e-01 0.077797820
## 70  3.638974e-01 2.936081e-01 1.936389e-01 3.228956e-01 0.126780860
## 71  3.439124e+00 5.473414e+00 2.848879e+00 2.191001e+00 1.160046677
## 72  8.506455e-01 5.236010e-01 1.103103e-01 2.103763e-03 0.007953058
## 73  4.013195e+00 2.516021e+00 1.796287e+00 1.205837e-01 1.976794022
## 74  8.925124e-02 5.152613e+00 1.247605e+00 1.078513e+00 1.643239634
## 75  5.859407e-01 5.380277e-01 4.650612e+00 7.095015e+00 8.094692190
## 76  2.309525e-03 3.482015e-01 1.794253e-01 1.452283e+00 0.046598581
## 77  2.606581e-01 3.312703e-01 6.461278e-03 2.716163e-01 0.562474280
## 78  2.919037e+00 5.526528e+00 6.891657e-01 1.248705e+00 0.101563779
## 79  1.519493e+00 9.478194e-02 2.895040e-04 7.442312e-01 0.073619755
## 80  1.848929e+00 1.044905e-01 1.970315e-02 1.393539e-01 0.630278828
## 81  5.043255e-03 2.406809e-01 2.129151e-02 2.933384e-01 1.688563128
## 82  6.625214e-01 1.923550e-01 1.907241e-07 5.114736e-02 0.137034868
```

```
## 83  1.079766e+00 1.838350e+00 2.374713e-02 1.125553e-01 0.084168080
## 84  2.790276e-01 4.034794e-01 2.531000e-01 5.124716e+00 0.070799478
## 85  1.793706e+00 7.090597e-03 2.250577e-01 5.481225e-02 0.565605796
## 86  3.348991e-01 1.497787e+00 1.365919e+00 1.465836e-01 0.061429211
## 87  9.030106e-01 7.221266e-01 2.106849e+00 5.931399e-04 1.816360362
## 88  9.195290e-02 1.202161e-01 7.407196e-01 2.725287e+00 2.330564406
## 89  2.774451e-03 4.701218e-02 7.733788e-01 1.128140e-01 0.338463263
## 90  7.388511e-05 4.673266e+00 1.028758e-02 1.359257e+00 0.197805941
## 91  3.209855e-02 8.423792e-01 3.794728e-02 5.392493e-02 2.066917364
## 92  7.459698e-01 9.686369e-01 3.704551e-03 8.837684e-01 0.007679820
## 93  4.921495e-03 3.308664e-02 1.817801e+00 5.214570e-01 0.873210514
## 94  5.414675e-01 6.762207e-02 2.362098e-02 3.853194e-03 0.865749134
## 95  1.092060e+00 1.640362e-03 4.061438e-01 2.319589e-01 0.004684129
## 96  1.441164e-01 1.111466e+00 2.361434e-01 2.181261e-01 0.308887380
## 97  8.697096e-01 4.804730e-01 6.895770e-02 8.723991e-01 3.531831893
## 98  2.758242e+00 1.439222e+00 2.276920e+00 2.740230e+00 0.366798622
## 99  1.880328e-01 1.625033e-01 2.573797e-02 5.520072e-02 3.495523893
## 100 2.591872e-01 1.646358e-01 1.859710e+00 3.696874e+00 2.696189324
##            Dim.6        Dim.7        Dim.8
## 1   0.2220063542 2.519854e-01 3.776440e-02
## 2   0.0557557015 1.523832e-01 2.993976e-01
## 3   0.1454199199 1.747187e+00 2.299933e-01
## 4   0.1870654111 6.225752e-01 6.968402e-03
## 5   5.9681693519 4.143799e+00 8.636702e-01
## 6   1.1231190464 2.020571e+00 2.460046e+00
## 7   0.1619775857 7.280861e-01 1.161143e-02
## 8   0.0149088466 1.346185e-01 1.896203e-01
## 9   1.7161916240 2.479696e-01 1.484217e-01
## 10  0.2063684519 2.305728e-01 4.485283e-01
## 11  1.0372998839 3.162617e+00 1.483635e+00
## 12  1.5207436892 1.212795e+00 1.185231e-01
## 13  0.9205608021 1.317477e+00 1.025771e-04
## 14  0.2420129502 4.020948e-02 2.881139e+00
## 15  5.3457919401 1.247019e-01 1.069720e-01
## 16  0.7514277534 1.939874e-01 7.718693e-02
## 17  3.3014922721 6.447553e-01 1.846167e+00
## 18  0.1055653169 2.012067e-02 1.796844e+00
## 19  0.0935167672 7.003963e-02 2.482197e+00
## 20  0.1388726654 2.515900e-01 6.375027e-01
## 21  0.4378640470 3.376771e-02 2.098161e-02
## 22  0.1868219566 6.718665e-04 2.656493e+00
## 23  0.0627365082 1.477638e-01 4.709521e-01
## 24  0.0358052073 2.617126e-01 1.756608e+00
## 25  3.0467111645 1.595806e+00 2.997623e-02
## 26  1.0316772229 4.465616e-01 7.509386e-01
## 27  0.0757906461 2.363872e-01 2.705690e-01
## 28  0.1806774389 5.937185e-01 8.027241e-01
## 29  1.1150904314 3.031223e-02 2.381075e+01
## 30  0.1717629668 6.393382e-02 3.474566e-01
## 31  0.0466285333 8.100514e-01 1.239256e-01
## 32  1.0539689657 2.107207e-03 1.298235e+00
## 33  0.3255759470 1.249738e+00 1.768415e-01
## 34  0.4707592126 2.579259e+00 5.787151e-01
## 35  0.2166030477 4.353186e-01 1.967784e+00
```

```
## 36  0.1317681610 1.291478e+00 6.189462e-02
## 37  0.4064653694 2.653296e-01 1.908234e-01
## 38  4.5256692510 1.948537e-01 8.691816e-02
## 39  0.0777408642 1.547476e-01 4.074024e-03
## 40  0.0652972984 2.668460e+00 2.479137e+00
## 41  0.0009041962 2.991178e+00 3.084543e-01
## 42  0.2452035877 2.421680e-01 7.481115e-01
## 43  1.2385674656 4.584227e-01 8.332496e-01
## 44  0.0818333886 8.163216e-03 3.460734e-02
## 45  0.0806614629 1.212181e+00 7.760781e-01
## 46  0.1802601275 3.260551e-02 4.014853e-04
## 47  0.5568415817 9.001690e+00 4.425989e+00
## 48  0.0346982571 3.504899e-01 4.288633e-02
## 49  1.4415637946 2.821410e+00 1.634188e+00
## 50  0.9553822399 1.216161e-02 2.112444e-01
## 51  0.0306496479 4.410685e-01 5.053044e-02
## 52  0.3595037233 1.745137e-02 6.541761e-02
## 53  0.1274359339 6.066677e+00 4.567313e-01
## 54  0.3722615776 1.610416e+00 2.234418e-01
## 55  4.2349407138 5.948559e-01 4.182186e-02
## 56  0.0261953059 1.306042e+00 8.507807e-02
## 57  0.1196965459 2.177972e-01 4.389549e-01
## 58  0.3499804019 3.397471e-02 4.830302e-02
## 59  0.0174332904 3.631886e+00 2.546861e+00
## 60  0.2581756234 2.785543e-02 4.106535e-02
## 61  0.0505569475 6.678806e-01 2.818478e+00
## 62  0.0422969199 1.428790e-01 5.021549e-01
## 63  1.2086688925 1.478511e-02 4.293910e-01
## 64  0.1259298113 3.235757e-01 1.402645e+00
## 65  0.0468423360 5.757506e-01 5.195268e-01
## 66  0.3951945076 1.104775e+00 1.725387e+00
## 67  0.1266435932 1.814734e-01 6.789529e-02
## 68  0.5207321376 1.808496e-01 1.487875e-01
## 69  1.2412215689 2.001865e+00 2.416348e-01
## 70  0.1431755346 1.986438e-01 4.737782e-01
## 71  0.0581224181 3.162024e-01 5.150289e-01
## 72  0.3592614876 2.204628e-01 7.798546e-02
## 73  1.7542975347 1.042695e+01 3.539555e+00
## 74  5.9625496176 1.989309e-01 1.427872e-01
## 75  0.8666319284 3.952476e-01 8.193691e-03
## 76  2.7360828573 2.940070e-04 4.146980e-01
## 77  0.4458801688 5.129966e-01 3.836385e+00
## 78  2.9258370479 1.984614e+00 2.060913e+00
## 79  0.0710892368 8.150333e-04 7.119005e-03
## 80  0.0074853020 1.402173e-01 6.977655e-03
## 81  0.1993064132 2.716726e+00 1.054738e+00
## 82  0.0010322389 6.775472e-02 1.344504e-02
## 83  0.4373575400 5.798721e-01 1.071853e-01
## 84  0.0045369685 1.071544e-01 5.678521e-02
## 85  0.0667080799 1.028910e+00 6.862371e-01
## 86  0.1429899898 3.199503e-02 1.891149e+00
## 87  1.5825412467 3.693973e-03 5.895677e-01
## 88  0.3818646679 1.843027e+00 1.988020e-01
## 89  0.8975040466 8.987143e-03 8.655632e-03
```

```
## 90  3.8443430204 7.083313e-01 1.233516e-02
## 91  0.0275169913 1.350454e-02 2.975601e-02
## 92  0.0083209496 3.729469e-01 4.280449e-01
## 93  2.4890460623 1.686709e+00 2.965493e+00
## 94  7.1507639261 2.915443e+00 2.345168e+00
## 95  0.4800855298 7.124861e-01 7.341244e-03
## 96  3.7746017720 3.083427e-02 8.862479e-01
## 97  5.0305493800 4.162415e-02 9.941210e-01
## 98  3.9915100914 5.169350e-01 6.095581e-01
## 99  0.3947625529 1.544497e+00 1.461603e-01
## 100 1.3702892451 3.020845e+00 4.379134e-03
```

4. Interpretation

```r
library(ggbiplot)
```

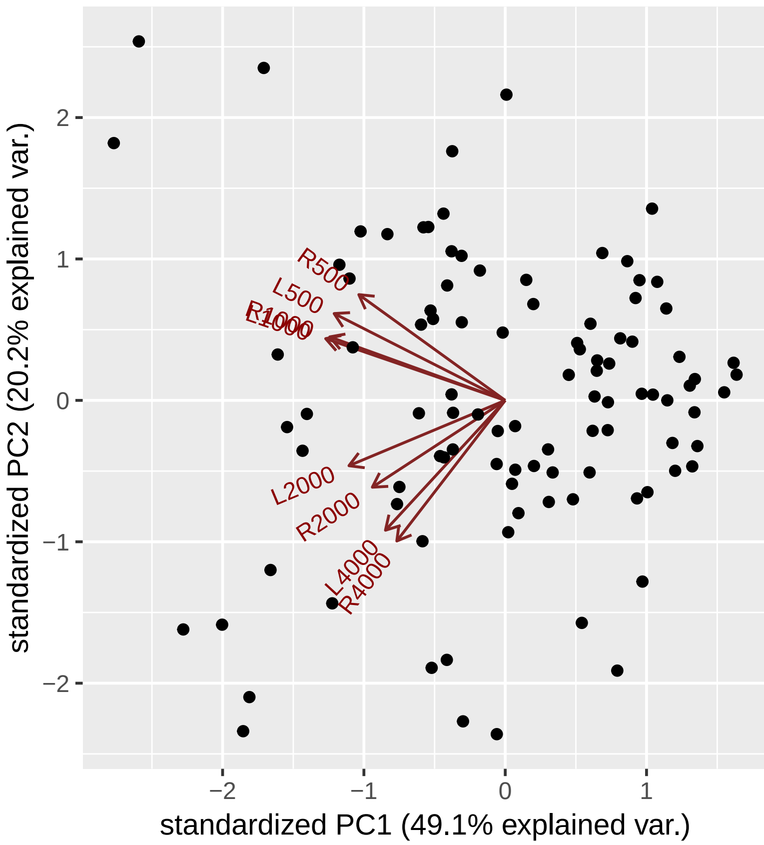
```
## Loading required package: plyr
```

```
## Loading required package: scales
```

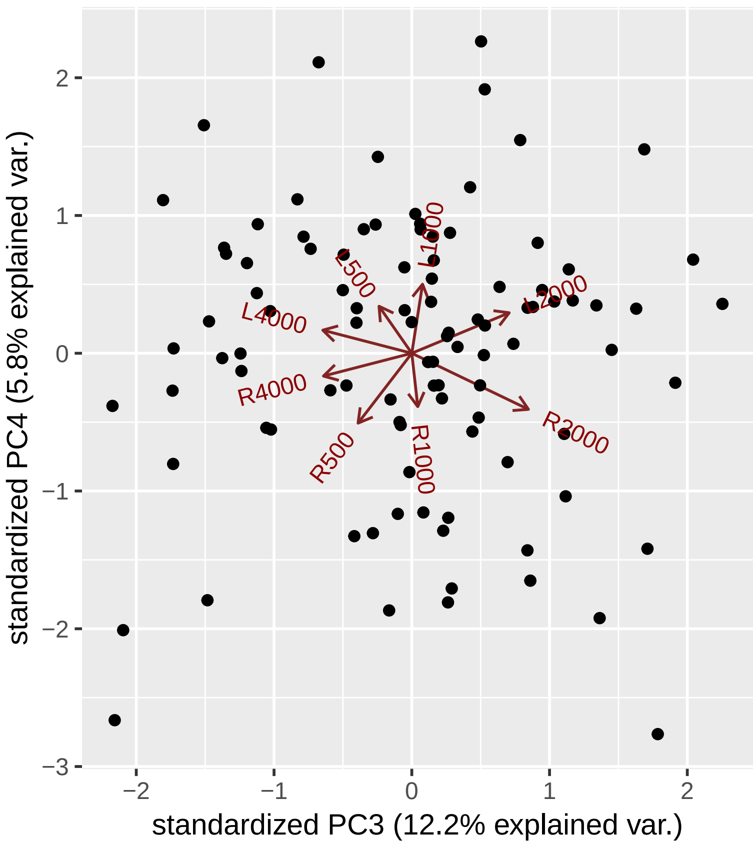
```
## Loading required package: grid
```

```r
library(scales)
library(plyr)

ggbiplot(prin_comp)
```




```r
ggbiplot(prin_comp, choices = c(3,4))
```



first plot shows two groupings of dosage amount independant of the drug

second plot barely shows any relevant information as there is barely any group seperation -> was to be expected since they explain small portion of the variance only

1. Problem Statement Task 2:

exploratory factor analysis: • explain the correlation structure among observed variables • try to find underlying dimensions that can explain the observed correlations • example: the correlation between scores on mathematics, statis- tics and physics exams can be explained because they all measure somehow quantitative intelligence

```
1. State the problem

2. Descriptive Statistics (to check data, to find outliers)

3. Test (or at least state) the assumptions of the method, if any

4. Conduct the method (describe in more detail the "best" approach you have found)

5. Interpret the solution

6. Compare the results briefly with alternative solutions, if any

7. Conclusion
```

Descriptive Statistics

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:scales':
##
##     alpha, rescale

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
corr <-read.delim("data/screening.txt", header = TRUE, sep="",dec = ".", skipNul = FALSE)
corr <- subset(corr, select = -c(X_name_))

m <- matrix(NA,20,20)
m[lower.tri(m,diag=TRUE)] <- 1:10

makeSymm <- function(m) {
   m[upper.tri(m)] <- t(m)[upper.tri(m)]
   return(m)
}

corr <- makeSymm(corr)
```

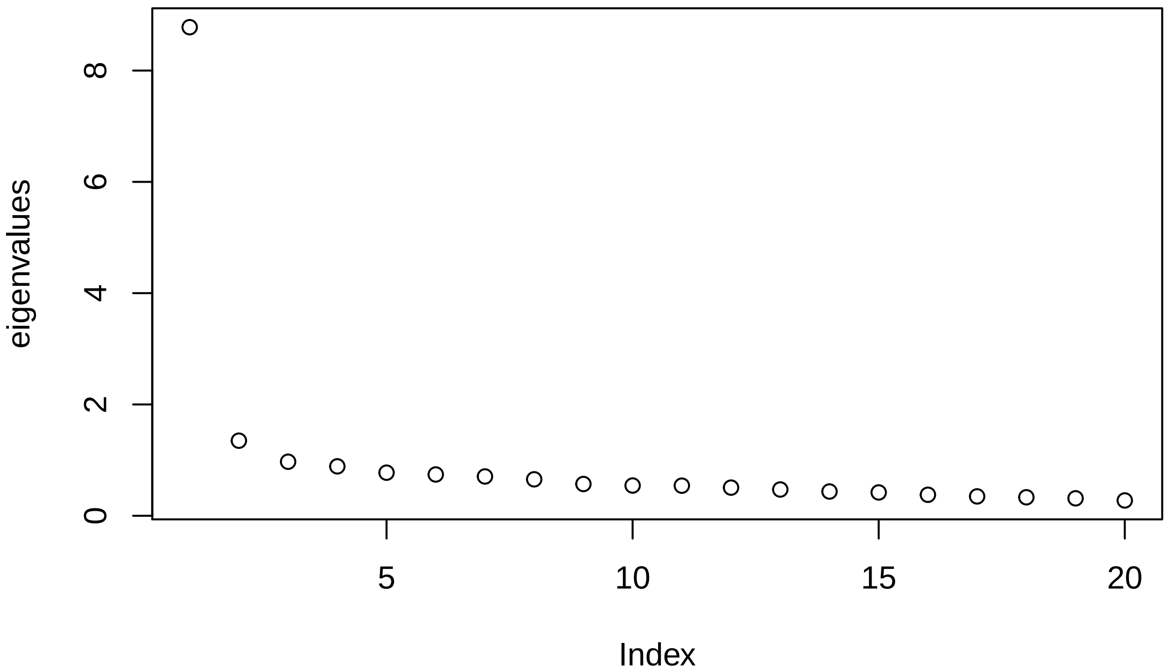
Assumptions of Methods

assuming standardized data and factors + uncorrelated factors

Method

compute eigenvalues and check how many factors should be extracted

```
eigenvalues <- eigen(corr)$values
plot(eigenvalues)
```



seems like the first component would be sufficient

```r
# set chosen number of factors
n <- 2
n_var <- 20

corr_smc <- (1 - 1 / diag(solve(corr)))
diag(corr) <- corr_smc

min.error <- .001
com.iter <- c()
h2 <- sum(diag(solve(corr)))
error <- h2

corr_eigen <- eigen(corr) # Get the eigenvalues and eigenvectors of R
  est <- if(n==1) sqrt(corr_eigen$values[1])  else diag(sqrt(corr_eigen$values[1:n]))
  lambda <- as.matrix(corr_eigen$vectors[,1:n]) %*% est
  while (error > min.error) {
  corr_eigen <- eigen(corr)

  # The lambda object is updated upon each iteration using new estimates of the communality
  est <- if(n==1) sqrt(corr_eigen$values[1])  else diag(sqrt(corr_eigen$values[1:n]))
  lambda <- as.matrix(corr_eigen$vectors[,1:n]) %*% est

  # R - Psi is then found by multiplying the lambda matrix by its transpose
  corr_mod <- lambda %*% t(lambda)
  corr_mod_diag <- diag(corr_mod) # The diagonal of R - Psi is the new communality estimate

  # The sum of the new estimate is taken and compared with the previous estimate. If the
  # difference is less than the error threshold the loop stops
  h2_new <- sum(corr_mod_diag)
  error <- abs(h2 - h2_new)
```

```r
  # If the difference between the previous and new estimate is not below the threshold, replace
  # the new estimate with the previous
  h2 <- h2_new

  # Store the iteration value (the sum of the estimate) and replace the diagonal of R with the
  # diagonal of R - Psi found previously
  com.iter <- append(com.iter, h2_new)
  diag(corr) <- corr_mod_diag
  }

h2 <- rowSums(lambda^2)
u2 <- 1 - h2
com <- rowSums(lambda^2)^2 / rowSums(lambda^4)

iter.fa.loadings <- data.frame(cbind(round(lambda,2), round(h2, 2), round(u2, 3), round(com, 2)))

cnames <- paste("Factor", as.character(c(1:n)))
colnames(iter.fa.loadings) <- c(cnames, 'h2', 'u2', 'com')


prop.var <- corr_eigen$values[1:n] / sum(diag(solve(corr)))
var.cumulative <- corr_eigen$values / n_var

factor.var <- data.frame(rbind(round(prop.var[1:n], 2), round(var.cumulative[1:n], 2)))
rownames(factor.var) <- c('Proportion Explained', 'Cumulative Variance')
cnames <- paste("Factor", as.character(c(1:n)))
colnames(factor.var) <- cnames
factor.var
```

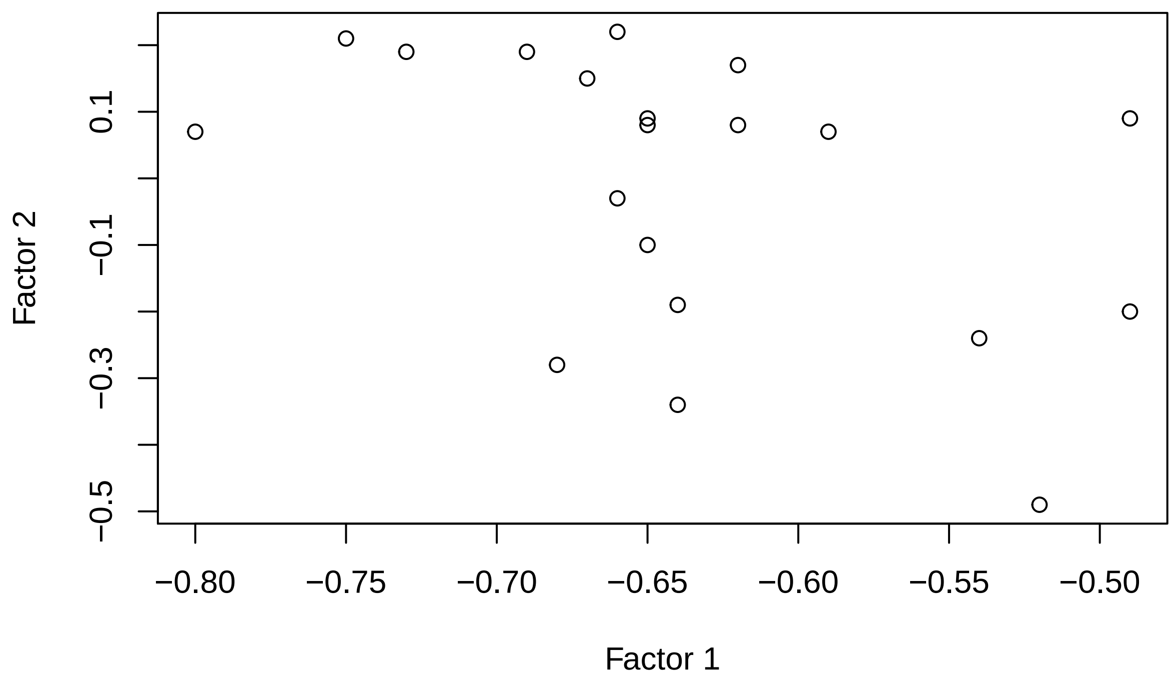
```
##                      Factor 1 Factor 2
## Proportion Explained    -0.01     0.00
## Cumulative Variance      0.41     0.04
```

```r
iter.fa.res <- list(iter.fa.loadings, factor.var)
iter.fa.res
```

```
## [[1]]
##    Factor 1 Factor 2   h2    u2  com
## 1     -0.62     0.08 0.39 0.609 1.03
## 2     -0.62     0.17 0.41 0.589 1.15
## 3     -0.54    -0.24 0.35 0.653 1.37
## 4     -0.65    -0.10 0.44 0.564 1.04
## 5     -0.52    -0.49 0.50 0.498 1.99
## 6     -0.49    -0.20 0.28 0.718 1.31
## 7     -0.68    -0.28 0.55 0.453 1.34
## 8     -0.69     0.19 0.51 0.487 1.15
## 9     -0.66     0.22 0.48 0.520 1.22
## 10    -0.49     0.09 0.25 0.748 1.07
## 11    -0.73     0.19 0.58 0.425 1.14
## 12    -0.65     0.09 0.43 0.570 1.04
## 13    -0.59     0.07 0.36 0.644 1.03
## 14    -0.64    -0.19 0.45 0.552 1.17
## 15    -0.80     0.07 0.64 0.359 1.02
```

```
## 16    -0.75     0.21 0.61 0.393 1.16
## 17    -0.64    -0.34 0.52 0.475 1.51
## 18    -0.67     0.15 0.47 0.533 1.10
## 19    -0.65     0.08 0.43 0.570 1.03
## 20    -0.66    -0.03 0.44 0.562 1.00
##
## [[2]]
##                      Factor 1 Factor 2
## Proportion Explained    -0.01     0.00
## Cumulative Variance      0.41     0.04
```

```r
plot(x=iter.fa.loadings[1:n])
```



5. Interpretation of Solution