PCA_V3

Howard Nguyen

2022-10-16

Loading libraries

```
library(psych);library(rela);library(MASS);library(parallel)
```

```
#  Return decimal values
options(scipen=999)  # report decimal values
```

```
pca = read.delim(file = "ch7ex1.dat", header=TRUE,sep="\t")
colnames(pca) = c("IRLETT", "FIGREL", "IRSYMB","CULTFR","RAVEN")
pca = as.matrix(pca)      # specify as matrix for use in other packages
head(pca,10)
```

```
##       IRLETT FIGREL IRSYMB CULTFR  RAVEN
##  [1,] 57.160 73.811 62.876 72.228 66.672
##  [2,] 20.006 38.096 17.148 43.059 38.892
##  [3,] 35.725 69.049 61.447 63.894 61.116
##  [4,] 15.719 30.953 11.432 30.558 16.668
##  [5,] 11.432 42.858 18.577 45.837 22.224
##  [6,] 24.293 61.906 30.009 51.393 44.448
##  [7,] 10.003 50.001 27.151 40.281 44.448
##  [8,] 34.296 33.334 32.867 47.226 22.224
##  [9,] 28.580 45.239 41.441 47.226 27.780
## [10,] 38.583 69.049 51.444 56.949 83.340
```

```
tail(pca,10)
```

```
##        IRLETT FIGREL IRSYMB CULTFR  RAVEN
## [152,] 21.435 35.715 30.009 50.004 38.892
## [153,] 18.577 40.477 12.861 37.503 27.780
## [154,] 48.586 54.763 51.444 54.171 66.672
## [155,] 47.157 64.287 55.731 61.116 55.560
## [156,] 42.870 64.287 55.731 58.338 72.228
## [157,] 30.009 50.001 40.012 45.837 55.560
## [158,] 27.151 50.001 37.154 52.782 44.448
## [159,] 52.873 57.144 57.160 68.061 66.672
## [160,] 47.157 57.144 62.876 61.116 44.448
## [161,] 40.012 59.525 48.586 55.560 50.004
```

```
# view dataset in table format
r_table = read.table(file = "ch7ex1.dat", header=TRUE,sep="\t")
```

Correlation matrix is computed with the following R command:

```
pcacor <- cor(pca)  # Compute correlation matrix
pcacor
```

```
##           IRLETT    FIGREL    IRSYMB    CULTFR     RAVEN
## IRLETT 1.0000000 0.6506553 0.8532134 0.6812657 0.6158707
## FIGREL 0.6506553 1.0000000 0.7185869 0.7613804 0.6357581
## IRSYMB 0.8532134 0.7185869 1.0000000 0.7652237 0.6762137
## CULTFR 0.6812657 0.7613804 0.7652237 1.0000000 0.6389802
## RAVEN  0.6158707 0.6357581 0.6762137 0.6389802 1.0000000
```

Variance-covariance matrix is computed with the following R command:

```
pcacov <- cov(pca)
pcacov
```

```
##          IRLETT   FIGREL   IRSYMB   CULTFR    RAVEN
## IRLETT 218.0572 155.1517 220.9215 127.0058 183.4952
## FIGREL 155.1517 260.7598 203.4674 155.2186 207.1391
## IRSYMB 220.9215 203.4674 307.4611 169.3969 239.2373
## CULTFR 127.0058 155.2186 169.3969 159.3836 162.7643
## RAVEN  183.4952 207.1391 239.2373 162.7643 407.0987
```

convert the covariance matrix to a correlation matrix using the cov2cor() function. The R command is as follows:

```
convert <- cov2cor(pcacor)
convert
```

```
##           IRLETT    FIGREL    IRSYMB    CULTFR     RAVEN
## IRLETT 1.0000000 0.6506553 0.8532134 0.6812657 0.6158707
## FIGREL 0.6506553 1.0000000 0.7185869 0.7613804 0.6357581
## IRSYMB 0.8532134 0.7185869 1.0000000 0.7652237 0.6762137
## CULTFR 0.6812657 0.7613804 0.7652237 1.0000000 0.6389802
## RAVEN  0.6158707 0.6357581 0.6762137 0.6389802 1.0000000
```

check the statistical significance of the bivariate correlations using the psych package and the corr.p() function. The R commands are as follows:

```
out <- corr.p(cor(pca), 161, alpha = .05)
print(out)
```

```
## Call:corr.p(r = cor(pca), n = 161, alpha = 0.05)
## Correlation matrix
##        IRLETT FIGREL IRSYMB CULTFR RAVEN
## IRLETT   1.00   0.65   0.85   0.68  0.62
## FIGREL   0.65   1.00   0.72   0.76  0.64
## IRSYMB   0.85   0.72   1.00   0.77  0.68
## CULTFR   0.68   0.76   0.77   1.00  0.64
## RAVEN    0.62   0.64   0.68   0.64  1.00
## Sample Size
## [1] 161
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##        IRLETT FIGREL IRSYMB CULTFR RAVEN
## IRLETT      0      0      0      0     0
## FIGREL      0      0      0      0     0
## IRSYMB      0      0      0      0     0
## CULTFR      0      0      0      0     0
## RAVEN       0      0      0      0     0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

Assumptions:

We are now ready to determine if we can proceed with a PCA. There are three assumptions we should always check: (1) sphericity, (2) sample adequacy, and (3) positive determinant of the matrix. The Bartlett chi-square tests whether the matrix displays sphericity—that is, an identity matrix. An identity matrix would have 1s on the diagonal and 0s on the off diagonal; thus, no correlation exists. The Bartlett test needs to be statistically significant to proceed—that is, sufficient correlation must exist in the matrix. The KMO test ranges from 0 to 1, with values closer to 1 indicating sample size adequacy. The determinant of the correlation matrix needs to be positive, which indicates that we can extract variance.

The Bartlett and KMO tests are in the rela package and are computed using the paf() function.

```
paf.pca <- paf(pca, eigcrit = 1, convcrit = .001)
summary(paf.pca)
```

```
## $KMO
## [1] 0.85876
##
## $MSA
##             MSA
## IRLETT 0.82285
## FIGREL 0.88671
## IRSYMB 0.80211
## CULTFR 0.87593
## RAVEN  0.94344
##
## $Bartlett
## [1] 614.15
##
```

```
## $Communalities
##        Initial Communalities Final Extraction
## IRLETT               0.73264          0.70657
## FIGREL               0.64069          0.67328
## IRSYMB               0.80606          0.85510
## CULTFR               0.68467          0.72786
## RAVEN                0.51579          0.55627
##
## $Factor.Loadings
##           [,1]
## IRLETT 0.84058
## FIGREL 0.82054
## IRSYMB 0.92472
## CULTFR 0.85315
## RAVEN  0.74583
##
## $RMS
## [1] 0.039232
```

The KMO test is close to 1 (KMO = .86), so we would conclude that n = 161 with 5 variables is an adequate sample size. Recall, many multivariate statistics books cite a 20:1 rule of thumb (5 variables $\times$ 20 = 100 subjects). The reported Bartlett chi-square of 614.15 is not indicated with a p value; therefore, we must run the following R command to determine statistical significance.

```r
# Test significance of the Bartlett test using correlation matrix
cortest.bartlett(pcacor, n = 161)
```

```
## $chisq
## [1] 614.15
##
## $p.value
## [1] 0.00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
##
## $df
## [1] 10
```

The Bartlett 2 = 614.15, df = 10, p < .00001 (the scientific notation overrides the printing of decimal values when extreme). Our final concern is the determinant of the matrix. The determinant is positive (.02). The R command is as follows:

```r
det(pcacor)
```

```
## [1] 0.020255
```

We have now satisfied our three asssumptions for conducting a PCA.

Number of Components

The PCA is computed using the psych package and the principal() function (not the fa() function). The default R command setting is given, which provides for one component and no component scores.

```r
pcmodel1 <- principal(pcacor, rotate = "none")  # default with 1 component and no scores
pcmodel1
```

```
## Principal Components Analysis
## Call: principal(r = pcacor, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##         PC1   h2   u2 com
## IRLETT 0.87 0.76 0.24   1
## FIGREL 0.86 0.75 0.25   1
## IRSYMB 0.92 0.85 0.15   1
## CULTFR 0.88 0.78 0.22   1
## RAVEN  0.81 0.66 0.34   1
##
##                 PC1
## SS loadings    3.80
## Proportion Var 0.76
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##
## Fit based upon off diagonal values = 0.99
```

We can interpret this initial output as follows. The SS loadings is equal to 3.80, which is the eigenvalue for the single principal component. The proportion variance equal to .76 is the average of the h2 values (Σh2/m). The eigenvalue is the sum of the h2 values; therefore, Σh2/m = 3.80/5 = .76! That leaves 24% unexplained variance. This could be due to another principal component or residual error variance.

To view more eigenvalues that represent 100% of the variance in the correlation matrix, we can extend the number of components and use the following R command:

```r
pcmodel2 <- principal(pcacor, nfactors = 5, rotate = "none")
pcmodel2
```

```
## Principal Components Analysis
## Call: principal(r = pcacor, nfactors = 5, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##         PC1   PC2   PC3   PC4   PC5 h2                   u2 com
## IRLETT 0.87 -0.39  0.17  0.08  0.21  1 0.00000000000000067 1.6
## FIGREL 0.86  0.19 -0.35  0.32  0.00  1 0.00000000000000289 1.7
## IRSYMB 0.92 -0.24  0.08 -0.02 -0.29  1 0.00000000000000200 1.4
## CULTFR 0.88  0.07 -0.28 -0.36  0.07  1 0.00000000000000167 1.6
## RAVEN  0.81  0.42  0.41 -0.01  0.02  1 0.00000000000000122 2.0
##
##                        PC1  PC2  PC3  PC4  PC5
## SS loadings           3.80 0.43 0.40 0.24 0.13
## Proportion Var        0.76 0.09 0.08 0.05 0.03
## Cumulative Var        0.76 0.85 0.93 0.97 1.00
## Proportion Explained  0.76 0.09 0.08 0.05 0.03
## Cumulative Proportion 0.76 0.85 0.93 0.97 1.00
##
## Mean item complexity =  1.7
```

```
## Test of the hypothesis that 5 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0
##
## Fit based upon off diagonal values = 1
```

The eigenvalues for the 5 principal components are given in descending order: PC1 = 3.8 (76%), PC2 = .43 (9%), PC3 = .40 (8%), PC4 = .24 (5%), and PC5 = .13 (3%). The sum of the eigenvalues explained variance equals 100%. (Note: There is rounding error in the percents listed using two decimal places). The cumulative variance, however, indicates the incremental explained variance from PC1 to PC5 that sums to 100%. The h2 (variale explained variance) is now 1.0, although u2 (residual variance) does indicate a very small amount of residual error.

A check of the Cronbach's alpha reliability coefficient indicates high internal consistency of response (= .92); so it does not affect the PCA results.

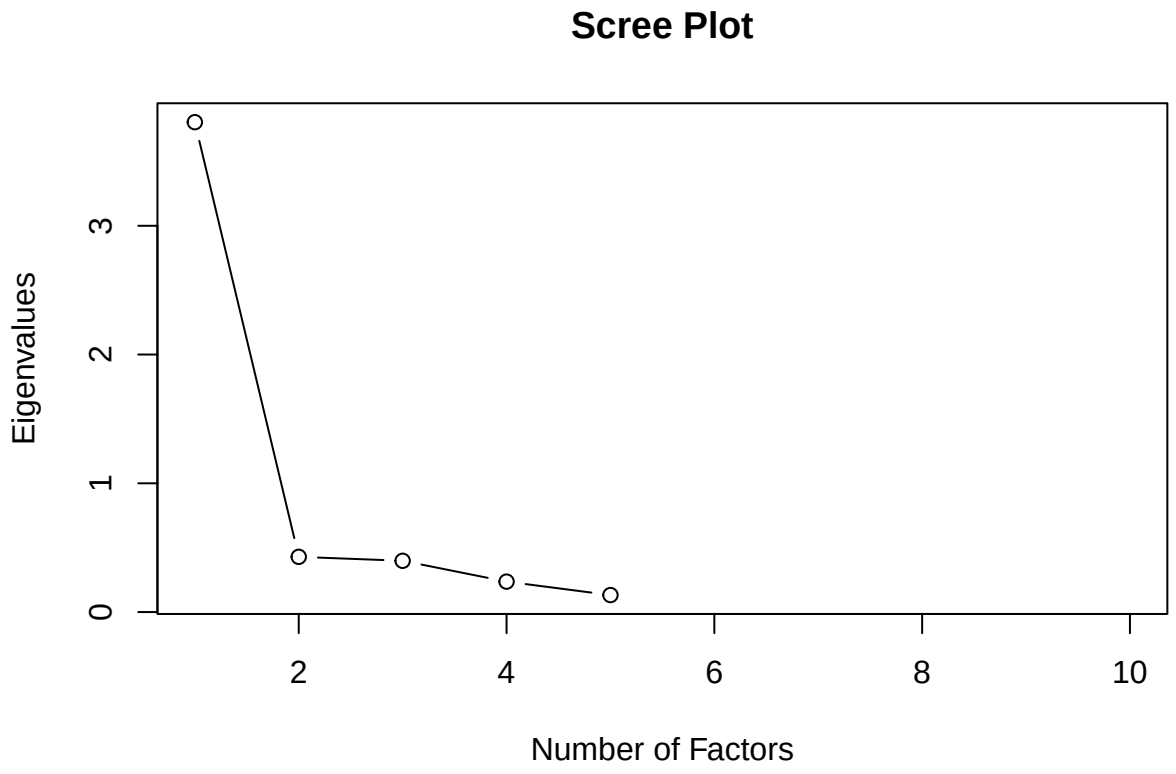
```
alpha(pcacor)
```

```
##
## Reliability analysis
## Call: alpha(x = pcacor)
##
##   raw_alpha std.alpha G6(smc) average_r S/N median_r
##       0.92      0.92    0.91       0.7  12     0.68
##
##     95% confidence boundaries
##       lower alpha upper
## Feldt  0.71  0.92  0.99
##
##  Reliability if an item is dropped:
##        raw_alpha std.alpha G6(smc) average_r  S/N  var.r med.r
## IRLETT      0.90      0.90    0.88      0.70  9.3 0.0034  0.70
## FIGREL      0.91      0.91    0.89      0.71  9.6 0.0079  0.68
## IRSYMB      0.89      0.89    0.86      0.66  7.9 0.0027  0.64
## CULTFR      0.90      0.90    0.89      0.69  9.0 0.0075  0.66
## RAVEN       0.92      0.92    0.91      0.74 11.3 0.0052  0.74
##
##  Item statistics
##           r r.cor r.drop
## IRLETT 0.87  0.85   0.80
## FIGREL 0.86  0.82   0.78
## IRSYMB 0.92  0.92   0.87
## CULTFR 0.88  0.85   0.81
## RAVEN  0.82  0.74   0.72
```

Scree Plot

The scree plot is a very useful tool when deciding how many principal components are required to explain the variable correlation (covariance). The general rule is to select eigenvalues that are greater than 1.0. We already have seen the eigenvalues for the five principal components. The first component has an eigenvalue of 3.8, while all others were less than 1.0. We should see this when plotting the eigenvalues. The R command is as follows:

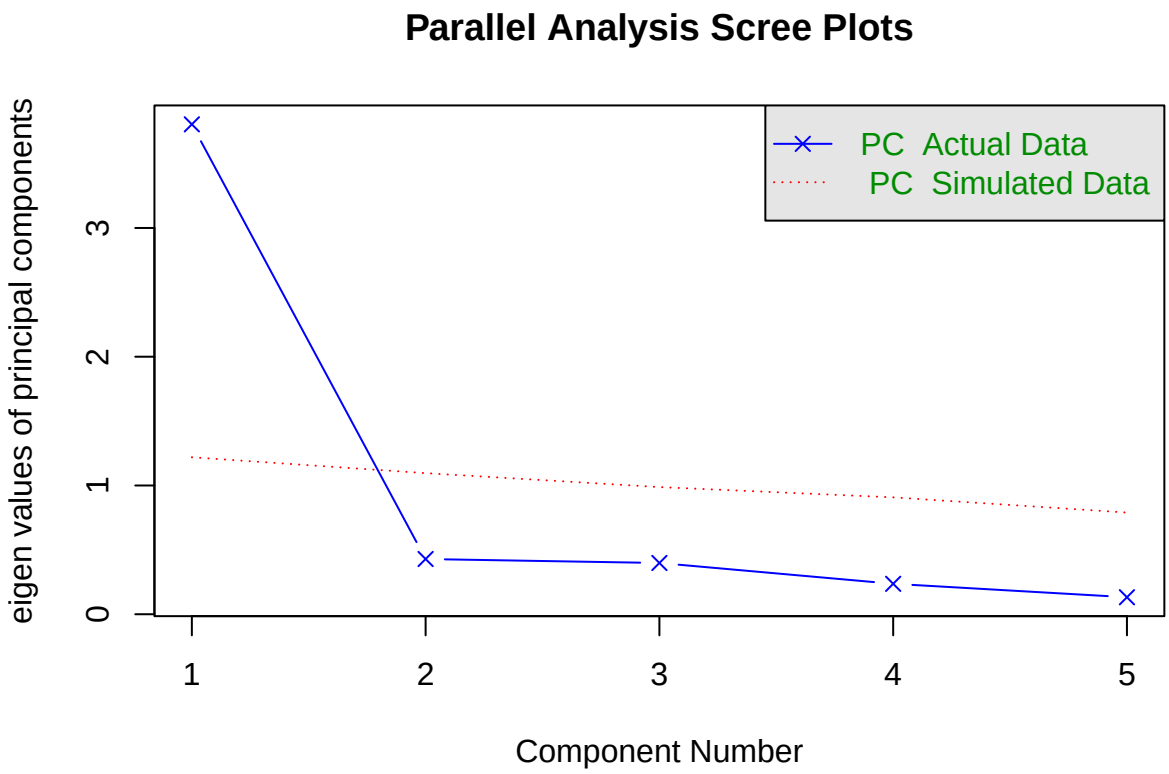
```
plot(pcmodel2$values, type = "b", xlim=c(1,10), main = "Scree Plot",
     xlab="Number of Factors", ylab="Eigenvalues")
```



Another type of scree plot is the parallel scree plot. The fa.parallel() function, however, now includes the arguments fm="pa" and fa="pc" for a principal components, rather than a factor, analysis. The parallel scree plot is given by the R command:

```
fa.parallel(pca, n.obs = 161, fm = "pa", fa = "pc")
```

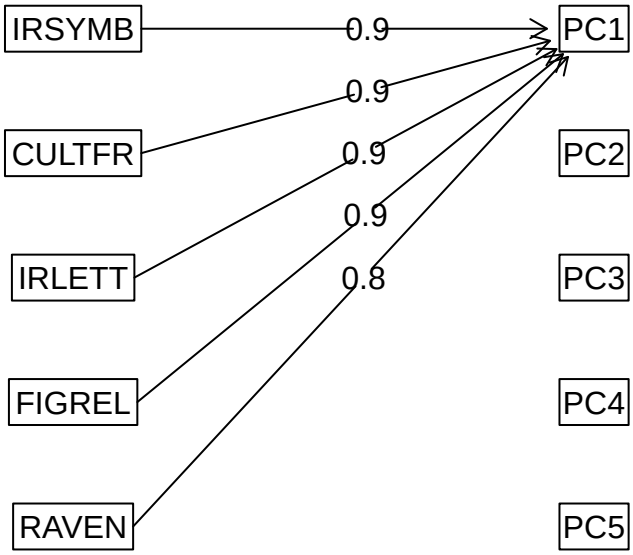
```
## Warning in fa.parallel(pca, n.obs = 161, fm = "pa", fa = "pc"): You specified
## the number of subjects, implying a correlation matrix, but do not have a
## correlation matrix, correlations found
```




```
## Parallel analysis suggests that the number of factors =  NA  and the number of components =  1
```

A plot of the five principal component model reveals graphically the component structure. The factor analysis loadings show high validity coefficients (.8, .9). These would be used to compute factor scores and are scaled accordingly

```
fa.diagram(pcmodel2)
```



Results indicated that a single component will summarize the five variable relations and yield 76% of the variable variance. The principal component equation to generate the scores is computed using the first set of weights. Yi = .87(IRLETT)+.86(FIGREL)+.92(IRSYMB)+.88(CULTFR)+.81(RAVEN)

compute scale score

We will need to first declare the PCA data set as a data frame. This is done so that the variable names can be used in the formula. Next, we compute the principal component scores using the weights in a formula:

```
pca2 <- data.frame(pca)  # converts pca matrix to pca2 data frame
attach(pca2)             # attach pca2 data frame to use variable names
```

```
pcscores = .87*IRLETT +.86*FIGREL +.92*IRSYMB + .88*CULTFR + .81*RAVEN
pcscores = sort(pcscores,decreasing=FALSE)
pcscores
```

```
##   [1]  66.213  68.637  72.503  76.810  78.953  79.551  84.867  87.248  89.191
##  [10]  89.709  89.991  90.550  91.205  95.636  96.006 104.952 105.258 107.413
##  [19] 107.830 108.742 109.930 109.967 110.924 113.393 113.514 116.469 117.148
##  [28] 117.347 118.309 118.697 119.498 120.276 121.047 122.097 122.233 122.366
##  [37] 126.032 127.904 129.690 129.967 130.087 130.378 130.877 131.183 132.103
##  [46] 132.916 134.013 135.338 137.440 139.057 139.215 139.842 140.661 140.929
##  [55] 141.955 144.532 145.171 147.253 147.606 148.133 148.303 148.620 149.132
##  [64] 149.196 149.239 149.481 149.523 149.746 150.122 151.098 151.901 152.478
##  [73] 152.803 154.594 156.700 157.472 158.693 163.247 163.804 164.015 165.270
```

```
##  [82] 165.957 167.372 170.332 173.946 174.577 175.472 177.328 179.502 180.581
##  [91] 183.193 183.211 183.255 183.743 188.457 188.846 190.460 190.623 191.260
## [100] 192.115 194.793 195.040 195.100 195.850 196.558 198.746 200.038 206.094
## [109] 207.970 209.585 210.206 210.906 212.438 215.468 216.833 218.474 219.792
## [118] 220.097 223.823 224.826 225.808 226.536 226.862 227.245 234.470 235.851
## [127] 237.801 238.369 238.440 239.741 243.945 244.612 246.372 246.485 247.534
## [136] 247.931 248.449 249.320 249.965 252.725 253.698 255.213 257.898 258.235
## [145] 258.359 261.629 263.630 263.670 264.505 267.520 272.076 274.278 284.499
## [154] 288.618 292.097 292.268 292.303 307.252 310.589 317.850 329.275
```

Once again, we find ourselves trying to make sense out of the scores. What does a 66.213 mean? We need to create a scale score that ranges from 0 to 100 for a more meaningful interpretation.

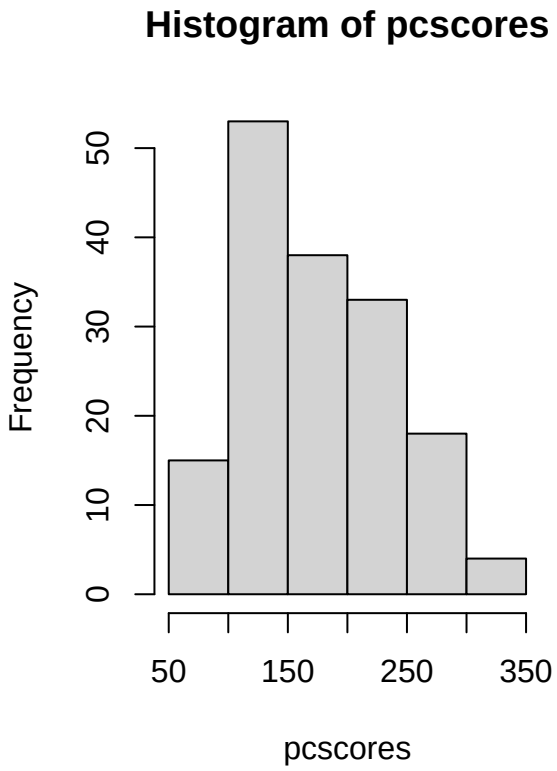
Use high and low scores in formula s = (100)/(329.275 -(-66.213)) = 100 / 395.488 = .25285 m = (0) - (66.213 * .25285) = 16.742

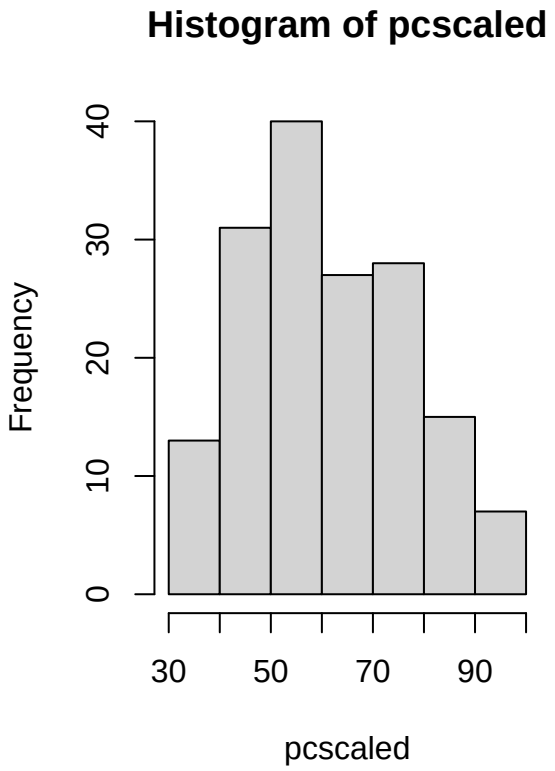
```
# compute scaled scores
pcscaled = 16.742 + (.25285 * pcscores) # compute scaled scores 0 to 100
round(pcscaled,2) # round numbers to 2 decimal places
```

```
##   [1]  33.48  34.10  35.07  36.16  36.71  36.86  38.20  38.80  39.29  39.42
##  [11]  39.50  39.64  39.80  40.92  41.02  43.28  43.36  43.90  44.01  44.24
##  [21]  44.54  44.55  44.79  45.41  45.44  46.19  46.36  46.41  46.66  46.75
##  [31]  46.96  47.15  47.35  47.61  47.65  47.68  48.61  49.08  49.53  49.60
##  [41]  49.63  49.71  49.83  49.91  50.14  50.35  50.63  50.96  51.49  51.90
##  [51]  51.94  52.10  52.31  52.38  52.64  53.29  53.45  53.97  54.06  54.20
##  [61]  54.24  54.32  54.45  54.47  54.48  54.54  54.55  54.61  54.70  54.95
##  [71]  55.15  55.30  55.38  55.83  56.36  56.56  56.87  58.02  58.16  58.21
##  [81]  58.53  58.70  59.06  59.81  60.72  60.88  61.11  61.58  62.13  62.40
##  [91]  63.06  63.07  63.08  63.20  64.39  64.49  64.90  64.94  65.10  65.32
## [101]  66.00  66.06  66.07  66.26  66.44  66.99  67.32  68.85  69.33  69.74
## [111]  69.89  70.07  70.46  71.22  71.57  71.98  72.32  72.39  73.34  73.59
## [121]  73.84  74.02  74.10  74.20  76.03  76.38  76.87  77.01  77.03  77.36
## [131]  78.42  78.59  79.04  79.07  79.33  79.43  79.56  79.78  79.95  80.64
## [141]  80.89  81.27  81.95  82.04  82.07  82.89  83.40  83.41  83.62  84.38
## [151]  85.54  86.09  88.68  89.72  90.60  90.64  90.65  94.43  95.27  97.11
## [161] 100.00
```

Once again, a graph of the principal component scores and the scaled scores show the equivalency. However, the scaled scores provide us with a meaningful interpretation. The five mental ability variables were reduced to a single component, which I will call Mental Ability. A person with a scaled score more than 50 would possess above average mental ability, while a person with a scaled score less than 50 would possess a lower than average mental abili y.

```
par(mfrow = c(1,2))  # formats a single graphic window with two histograms
hist(pcscores)
hist(pcscaled)
```





Reporting and Interpreting

Principal components analysis was conducted for 5 variables (IRLETT, FIGREL, IRSYMB, CULTFR, and RAVEN), which had statistically significant biv riate correlat ons. The assum tions for sphericity, sample adequacy, and determinant of the matrix were tested. The Bartlett chi-square test = 614.15 ($p < .00001$), which was statistically significant indicating that sufficient correlations were present in the matrix for analysis. The KMO test = .86. The KMO test is close to 1.0 (one), so we would conclude that n = 161 with 5 variables is an adequate sample size for the analysis. Finally, the determinant of the correlation matrix = .02, which is positive with the following eigenvalues in descending order for the 5 variables (3.80 0.43 0.40 0.24 0.13). The eigenvalues should sum to 5, the number of variables. PCA indicated a single unidimensional component with 76% variance explained. The scree plot indicated a single eigenvalue greater than one. Number of Components (page#5) indicates the standardized loadings (PC1), the commonality estimates (h2), and the residual estimates (u2). The h2 (explained variance) plus u2 (unexplained variance) equals one for each variable. The internal consistency reliability coefficient (Cronbach's alpha) indicated consistency of scores, that is, score reproducibility (= .92). Principal component scores would be computed using the following equation with the component variable weights:

```
pcmodel1
```

```
## Principal Components Analysis
## Call: principal(r = pcacor, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##         PC1   h2   u2 com
## IRLETT 0.87 0.76 0.24   1
```

```
## FIGREL 0.86 0.75 0.25   1
## IRSYMB 0.92 0.85 0.15   1
## CULTFR 0.88 0.78 0.22   1
## RAVEN  0.81 0.66 0.34   1
##
##                 PC1
## SS loadings    3.80
## Proportion Var 0.76
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##
## Fit based upon off diagonal values = 0.99
```

Yi=.87(IRLETT)+.86(FIGREL)+.92(IRSYMB)+.88(CULTFR)+.81(RAVEN)