In R there are several ways to do exploratory factor and principal components analysis. Best reference, and developer of the 'psych' package: William Revelle, see links inside R in documentation on the 'psych' package. Ch 6 of his forthcoming book is highly recommended. Also very good, Michael Friendly’s page, not specific to R:
http://www.psych.yorku.ca/lab/psy6140/fa/facplan.htm
My handout here is not intended to be a lecture handout, but a relatively quick reference for ‘how to’ in R.

Contents of this handout:
I. Preliminaries (test correlation matrix, find SMC, look for outliers)
II. Principal components analysis (two options, princomp or principal). Scree plots.
III. Factor analysis (‘factanal’ or ‘fa’)
IV. Other nifty things in the ‘psych’ package, including Very Simple Structure, parallel analysis (both help choose number of factors to fit), comparing factor analyses across samples or within sample, Kaiser-Meyer-Olin index of sampling adequacy, Cronbach’s alpha
V. Other nifty thing (from me). How to randomly split a sample in two to test sample separately.

> library(psych)  ## bring the psych package into R memory, for a lot of what is done below

I. Preliminaries (and how to do them) before diving into principal components or factor analysis

A. Test to see if your correlation matrix differs significantly from the identity matrix. You don’t want to be fitting just error. See section IV.A.1. below.
B. Do you have a reasonable set of measures, or do some items not belong in this analysis? Find the squared multiple correlations (SMC) of each variable with the others. Inspect for low values, read the items that have low SMC values, and decide whether to remove them. See section IV.A.2 below. If you are constructing a new scale, you will want to remove items after fitting a model also.
C. Look for outliers using Mahalanobis distances (D2):
   > outlier(asiq, plot=T, bad=10, na.rm=T)  ## in psych package
       In a large sample, ask it to flag more bad values than in a smaller sample. Also, Mahalanobis distances are supposed to be distributed as a chi-squared distribution, with df = number of variables going into the distance calculation. Can get some idea about how far out of your distribution the outliers are by looking at the p-values of the chisq distribution. For my 8 variable example below as follows:
   > qchisq(.01, 8, lower.tail=F)  ## p=.01, df=8, we want the upper tail
       [1] 20.09024
   ## this says that if an outlier has a distance over 20, it is in the upper 1% of distribution.
   > pchisq(15, 8, lower.tail=F)  ## this gives you the prob of a given chisq value
       [1] 0.05914546

II. Principal components
There are at least two ways to calculate principal components in R.
A. princomp — does principal components, yields eigenvalues.
       Minimal output, can’t control # of components??
   > pca2b=princomp(mat2,factors=2)  ## data are in mat2
   > summary(pca2b)
   Importance of components:
## look at scree plot, there are 2 ways to do this, note difference in scaling, the output immediately below is the square of the sd of the components shown above, the plot on the next page has the sd’s themselves.

```R
> screeplot(pca2b);
```

![Scree plot](image)
From inside the 'psych' package can also make scree plot. Notice different scaling from above. One is the square root of the other.

```r
> VSS.scree(mat2)
```

### From inside the 'psych' package can also make scree plot. Notice different scaling from above. One is the square root of the other.

```r
> VSS.scree(mat2)
```

---

**B. Can calculate principal components using 'principal' in 'psych' package**

```r
> pca2=principal(mat2, nfactors=2, rotate="varimax", scores=F)
> pca2  ## get the output from R
```

Principal Components Analysis

Call: principal(r = mat2, nfactors = 2, rotate = "varimax", scores = F)

Standardized loadings (pattern matrix) based upon correlation matrix

<table>
<thead>
<tr>
<th></th>
<th>RC1</th>
<th>RC2</th>
<th>h2</th>
<th>u2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Researchers_announce</td>
<td>0.16</td>
<td>0.90</td>
<td>0.84</td>
<td>0.157</td>
</tr>
<tr>
<td>Researchers_communicate_quickly</td>
<td>0.05</td>
<td>0.93</td>
<td>0.87</td>
<td>0.134</td>
</tr>
<tr>
<td>Researchers_pos_contribute</td>
<td>0.39</td>
<td>0.77</td>
<td>0.75</td>
<td>0.251</td>
</tr>
<tr>
<td>Researchers_available</td>
<td>0.25</td>
<td>0.88</td>
<td>0.84</td>
<td>0.156</td>
</tr>
<tr>
<td>Influence_worthwhile</td>
<td>0.92</td>
<td>0.23</td>
<td>0.90</td>
<td>0.097</td>
</tr>
<tr>
<td>Influence_benefits_community</td>
<td>0.92</td>
<td>0.18</td>
<td>0.89</td>
<td>0.113</td>
</tr>
<tr>
<td>Influence_important_topic</td>
<td>0.96</td>
<td>0.22</td>
<td>0.96</td>
<td>0.039</td>
</tr>
<tr>
<td>Influence_healthcare</td>
<td>0.93</td>
<td>0.14</td>
<td>0.89</td>
<td>0.110</td>
</tr>
</tbody>
</table>

RC1  RC2

SS loadings 3.74 3.21
Proportion Var 0.47 0.40
Cumulative Var 0.47 0.87
Proportion Explained 0.54 0.46
Cumulative Proportion 0.54 1.00

Test of the hypothesis that 2 components are sufficient.

The degrees of freedom for the null model are 28 and the objective function was 9.3
The degrees of freedom for the model are 13 and the objective function was 0.7
The total number of observations was 70 with MLE Chi Square = 44.89 with prob < 2.2e-05
Fit based upon off diagonal values = 1

## get more output
> pca2$loadings

Loadings:

<table>
<thead>
<tr>
<th>Factor</th>
<th>RC1</th>
<th>RC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Researchers_announce</td>
<td>0.157</td>
<td>0.905</td>
</tr>
<tr>
<td>Researchers_communicate_quickly</td>
<td>0.392</td>
<td>0.772</td>
</tr>
<tr>
<td>Researchers_pos_contribute</td>
<td>0.254</td>
<td>0.883</td>
</tr>
<tr>
<td>Influence_important_topic</td>
<td>0.956</td>
<td>0.216</td>
</tr>
<tr>
<td>Influence_healthcare</td>
<td>0.933</td>
<td>0.137</td>
</tr>
</tbody>
</table>

## See documentation for how to get residuals, scores, and other rotations. Notice that asking for the loadings stored, the ‘principal’ program in ‘psych’ package omits loadings below a low cutoff value.

### III. “Common factors” or true factor analysis

#### A. Can use `fa` in psych package

```r
> paf2 = fa(mat2, nfactors = 2, rotate = "varimax", SMC = T, symmetric = T, fm = "pa")
> paf2  ## ask for results
```

Factor Analysis using method = pa

Call: fa(r = mat2, nfactors = 2, rotate = "varimax", SMC = T, symmetric = T,
          fm = "pa")

Standardized loadings (pattern matrix) based upon correlation matrix

<table>
<thead>
<tr>
<th>Factor</th>
<th>PA1</th>
<th>PA2</th>
<th>h2</th>
<th>u2</th>
<th>com</th>
</tr>
</thead>
<tbody>
<tr>
<td>Researchers_announce</td>
<td>0.16</td>
<td>0.87</td>
<td>0.79</td>
<td>0.212</td>
<td>1.1</td>
</tr>
<tr>
<td>Researchers_communicate_quickly</td>
<td>0.06</td>
<td>0.90</td>
<td>0.82</td>
<td>0.178</td>
<td>1.0</td>
</tr>
<tr>
<td>Researchers_pos_contribute</td>
<td>0.38</td>
<td>0.72</td>
<td>0.67</td>
<td>0.334</td>
<td>1.5</td>
</tr>
<tr>
<td>Researchers_available</td>
<td>0.26</td>
<td>0.86</td>
<td>0.80</td>
<td>0.203</td>
<td>1.2</td>
</tr>
<tr>
<td>Influence_important_topic</td>
<td>0.90</td>
<td>0.23</td>
<td>0.87</td>
<td>0.131</td>
<td>1.1</td>
</tr>
<tr>
<td>Influence_benefits_community</td>
<td>0.90</td>
<td>0.19</td>
<td>0.84</td>
<td>0.160</td>
<td>1.1</td>
</tr>
<tr>
<td>Influence_important_topic</td>
<td>0.97</td>
<td>0.22</td>
<td>0.98</td>
<td>0.017</td>
<td>1.1</td>
</tr>
<tr>
<td>Influence_healthcare</td>
<td>0.90</td>
<td>0.15</td>
<td>0.84</td>
<td>0.160</td>
<td>1.1</td>
</tr>
</tbody>
</table>

## SS loadings 3.62 2.99
Proportion Var 0.45 0.37
Cumulative Var 0.45 0.83
Proportion Explained 0.55 0.45
Cumulative Proportion 0.55 1.00

Mean item complexity = 1.1

Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are 28 and the objective function was 9.3 with Chi Square of 609.43
The degrees of freedom for the model are 13 and the objective function was 0.44

The root mean square of the residuals (RMSR) is 0.02
The df corrected root mean square of the residuals is 0.02
The harmonic number of observations is 70 with the empirical chi square 1.13 with prob < 1
The total number of observations was 70 with MLE Chi Square = 28.15 with prob < 0.0086

Tucker Lewis Index of factoring reliability = 0.943
RMSEA index = 0.139 and the 90% confidence intervals are 0.062 0.195
BIC = -27.08
Fit based upon off diagonal values = 1

Measures of factor score adequacy

<table>
<thead>
<tr>
<th></th>
<th>PA1</th>
<th>PA2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation of scores with factors</td>
<td>0.99</td>
<td>0.96</td>
</tr>
<tr>
<td>Multiple R square of scores with factors</td>
<td>0.99</td>
<td>0.92</td>
</tr>
<tr>
<td>Minimum correlation of possible factor scores</td>
<td>0.98</td>
<td>0.85</td>
</tr>
</tbody>
</table>

## See documentation for other options for both rotation and factoring methods.

### B. Another option: factanal, which does maximum likelihood factor analysis

```r
> mlf2 = factanal(mat2, factors=2, rotation="varimax")
> mlf2; ## get R to show results

Call:
factanal(x = mat2, factors = 2, rotation = "varimax")

Uniquenesses:

<table>
<thead>
<tr>
<th>Researcher_announce</th>
<th>Researchers_communicate_quickly</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.198</td>
<td>0.172</td>
</tr>
<tr>
<td>Researchers_pos_contribute</td>
<td>Researchers_available</td>
</tr>
<tr>
<td>0.348</td>
<td>0.221</td>
</tr>
<tr>
<td>Influence_worthwhile</td>
<td>Influence_benefits_community</td>
</tr>
<tr>
<td>0.169</td>
<td>0.147</td>
</tr>
<tr>
<td>Influence_important_topic</td>
<td>Influence_healthcare</td>
</tr>
<tr>
<td>0.010</td>
<td>0.138</td>
</tr>
</tbody>
</table>

Loadings:

<table>
<thead>
<tr>
<th>Researcher_announce</th>
<th>Factor1</th>
<th>Factor2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.147</td>
<td>0.883</td>
<td></td>
</tr>
<tr>
<td>Researchers_communicate_quickly</td>
<td>0.908</td>
<td></td>
</tr>
<tr>
<td>0.364</td>
<td>0.721</td>
<td></td>
</tr>
<tr>
<td>Researchers_available</td>
<td>0.248</td>
<td>0.847</td>
</tr>
<tr>
<td>Influence_worthwhile</td>
<td>0.878</td>
<td>0.245</td>
</tr>
<tr>
<td>Influence_benefits_community</td>
<td>0.902</td>
<td>0.197</td>
</tr>
<tr>
<td>Influence_important_topic</td>
<td>0.968</td>
<td>0.228</td>
</tr>
<tr>
<td>Influence_healthcare</td>
<td>0.915</td>
<td>0.155</td>
</tr>
</tbody>
</table>

Factor1 Factor2

<table>
<thead>
<tr>
<th>SS loadings</th>
<th>Proportion Var</th>
<th>Cumulative Var</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.580</td>
<td>0.447</td>
<td>0.447</td>
</tr>
<tr>
<td>3.016</td>
<td>0.377</td>
<td>0.824</td>
</tr>
</tbody>
</table>

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 25.15 on 13 degrees of freedom.
The p-value is 0.0221

## See documentation for estimating factor scores, etc
IV. Other nifty things related to principal components or factor analysis in psych package

A. Bartlett’s test for a correlation matrix (is it identity matrix + error). You shouldn’t do factor analysis on a random matrix. Also known as Bartlett’s test of sphericity. You want the Bartlett test to have a small p-value.

```r
> cortest.bartlett(D2AxS[,26:31])  # example using columns 26:31 of my data
R was not square, finding R from data
$chisq
[1] 75.47375

$p.value
[1] 4.648804e-10

$df
[1] 15

> cortest.mat(D2AxS[,26:31]);  # also calculates Bartlett’s test
Bartlett's test of is R = I
Tests of correlation matrices
Call:cortest.mat(R1 = D2AxS[, 26:31])
Chi Square value 75.47 with df = 15 with probability < 4.6e-10
Warning message:
In cortest.mat(D2AxS[, 26:31]) :
  R1 matrix was not square, correlations found
> cortest.jennrich
compares matrices

> cortest.normal;  # differs but can use this to compare pairs of matrices, which is interesting to do if you have two samples tested on the same variables
```

B. Get the squared multiple correlations of each variable with all the others. Look at these to see if you should throw out some variables. Some say use a .30 (about 10% shared variance) criterion, but it is just a rule of thumb.

```r
> smc1= smc(D2AxS[,26:31]);  # a few columns of my data again
subnig      vta      acmb      amyg        bs      caud
0.7887606 0.6336429 0.9430341 0.8579995 0.6595345 0.9502663

When you use this, it is important to look at the content of the items and to think. You can also plot the cumulative distribution function of the squared multiple correlations and look at it to get a feel for whether some items don’t correlate very well with the others.

> plot.ecdf(smc1, main="Some Brain Areas, Sq mult corrs", xlab="sq mult corr")
```
C. **VSS, or Very Simple structure**, an aid to choosing number of factors.

VSS gives a lot of information, not just the VSS fit, which is also plotted. Complexity is the number of factors that each variable loads on. Choose the number of factors based on the maximum value of the VSS. Note, this gives a different answer for the data example below from the parallel analysis in the next section. This also included Velicer’s MAP function, which should be minimized.

```r
> VSS(asiq, n=8, rotate="varimax", fm="pa", plot=T)
```

*Different data set from earlier in this handout*

**Very Simple Structure**

Call: `vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm, n.obs = n.obs, plot = plot, title = title)`

VSS complexity 1 achieves a maximum of 0.64 with 2 factors
VSS complexity 2 achieves a maximum of 0.78 with 4 factors

The Velicer MAP achieves a minimum of 0.01 with 5 factors
BIC achieves a minimum of -3054.04 with 8 factors
Sample Size adjusted BIC achieves a minimum of -474.89 with 8 factors

**Statistics by number of factors**

<table>
<thead>
<tr>
<th>vss1</th>
<th>vss2</th>
<th>map</th>
<th>dof</th>
<th>chisq</th>
<th>prob</th>
<th>sqresid</th>
<th>RMSEA</th>
<th>BIC</th>
<th>SABIC</th>
<th>complex</th>
<th>eChisq</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.57</td>
<td>0.00</td>
<td>0.0207</td>
<td>1127</td>
<td>13383</td>
<td>0.0e+00</td>
<td>68</td>
<td>0.57</td>
<td>0.099</td>
<td>5451</td>
<td>9031</td>
</tr>
<tr>
<td>2</td>
<td>0.64</td>
<td>0.74</td>
<td>0.0130</td>
<td>1079</td>
<td>8950</td>
<td>0.0e+00</td>
<td>42</td>
<td>0.74</td>
<td>0.081</td>
<td>1356</td>
<td>4783</td>
</tr>
<tr>
<td>3</td>
<td>0.60</td>
<td>0.77</td>
<td>0.0087</td>
<td>1032</td>
<td>6313</td>
<td>0.0e+00</td>
<td>30</td>
<td>0.81</td>
<td>0.068</td>
<td>-950</td>
<td>2328</td>
</tr>
<tr>
<td>4</td>
<td>0.58</td>
<td>0.78</td>
<td>0.0073</td>
<td>986</td>
<td>4948</td>
<td>0.0e+00</td>
<td>25</td>
<td>0.84</td>
<td>0.060</td>
<td>-1991</td>
<td>1141</td>
</tr>
<tr>
<td>5</td>
<td>0.50</td>
<td>0.74</td>
<td>0.0069</td>
<td>941</td>
<td>4068</td>
<td>0.0e+00</td>
<td>22</td>
<td>0.86</td>
<td>0.055</td>
<td>-2555</td>
<td>434</td>
</tr>
<tr>
<td>6</td>
<td>0.50</td>
<td>0.74</td>
<td>0.0073</td>
<td>897</td>
<td>3618</td>
<td>2.6e-321</td>
<td>20</td>
<td>0.87</td>
<td>0.052</td>
<td>-2700</td>
<td>149</td>
</tr>
<tr>
<td>7</td>
<td>0.49</td>
<td>0.74</td>
<td>0.0076</td>
<td>854</td>
<td>3071</td>
<td>7.2e-247</td>
<td>19</td>
<td>0.88</td>
<td>0.048</td>
<td>-2940</td>
<td>-227</td>
</tr>
<tr>
<td>8</td>
<td>0.48</td>
<td>0.74</td>
<td>0.0076</td>
<td>812</td>
<td>2661</td>
<td>5.8e-195</td>
<td>17</td>
<td>0.89</td>
<td>0.045</td>
<td>-3054</td>
<td>-475</td>
</tr>
<tr>
<td>2.1</td>
<td>1753</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SRMR eCRMS eBIC
1 0.120 0.122 30389
2 0.078 0.082 8884
3 0.054 0.058 654
4 0.043 0.047 -1967
5 0.036 0.040 -3169
6 0.033 0.037 -3459
D. **Parallel analysis.** Choose the number of factors by simulating a random data set, and choosing the point where the eigenvalues of the real data fall below the simulated data.

```r
> pfa3 <- fa.parallel(asiq, fm="minres", fa="both")
```

Parallel analysis suggests that the number of factors = 10 and the number of components = 8

---

**Parallel Analysis Scree Plots**

---

**Very Simple Structure Fit**

---

**Parallel Analysis Scree Plots**
E. **Compare factor solutions.** Some writers say to “Factor the data by several different analytical procedures and hold sacred only those factors that appear across all the procedures used.” (Gorsuch, Factor Analysis, p. 330, 1983).

```r
> pca1=principal(asiq, nfactors=5, rotate="promax", scores=F)
> paf1=fa(asiq,nfactors=5,rotate="varimax",SMC=T,symmetric=T, fm="pa")
> factor.congruence(pca1,paf1); # compare princ comp and factor analysis
```

<table>
<thead>
<tr>
<th></th>
<th>PA1</th>
<th>PA3</th>
<th>PA2</th>
<th>PA4</th>
<th>PA5</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC3</td>
<td>-0.39</td>
<td>-0.98</td>
<td>0.12</td>
<td>0.02</td>
<td>-0.05</td>
</tr>
<tr>
<td>PC1</td>
<td>0.94</td>
<td>0.36</td>
<td>-0.09</td>
<td>0.00</td>
<td>0.24</td>
</tr>
<tr>
<td>PC2</td>
<td>-0.08</td>
<td>-0.03</td>
<td>0.97</td>
<td>-0.13</td>
<td>0.07</td>
</tr>
<tr>
<td>PC4</td>
<td>0.06</td>
<td>-0.04</td>
<td>-0.23</td>
<td>0.98</td>
<td>0.18</td>
</tr>
<tr>
<td>PC5</td>
<td>0.14</td>
<td>-0.04</td>
<td>0.01</td>
<td>-0.06</td>
<td>0.94</td>
</tr>
</tbody>
</table>

For this example, models are fit with 5 principal components or with 5 factors, and different rotations are applied. I have highlighted the diagonal elements, because the components/factors are not ordered the same.

Another way to do this is to use the solution from one set of data and apply it to another (for example, a random half of the sample).

```r
> predict.psych; (see documentation in R)
```

F. **Compare factor solutions by applying one analysis to another data set.** See documentation in psych package. Example with 2 data sets with 8 variables, make 2 principal components.

```r
## get pca from survey 1, apply to survey 2, then make correlations

> pca1=principal(survey1items, nfactors=2, rotate="varimax", scores=T);
> predpca2=predict(pca1, survey2items, survey1items);
> pca2=principal(survey2items, nfactors=2, rotate="varimax", scores=T);
> round(cor(pca2$scores,predpca2, use="pairwise.complete.obs"),2);
```

<table>
<thead>
<tr>
<th></th>
<th>RC1</th>
<th>RC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC1</td>
<td>0.16</td>
<td>0.98</td>
</tr>
<tr>
<td>RC2</td>
<td>0.99</td>
<td>0.17</td>
</tr>
</tbody>
</table>

### works in reverse too-- predict survey1 from survey2 pca

```r
> predpca1=predict(pca2, survey1items, survey2items);
> round(cor(pca1$scores, predpca1, use="pairwise.complete.obs"),3);
```

<table>
<thead>
<tr>
<th></th>
<th>RC1</th>
<th>RC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC1</td>
<td>-0.178</td>
<td>0.992</td>
</tr>
<tr>
<td>RC2</td>
<td>0.979</td>
<td>-0.124</td>
</tr>
</tbody>
</table>

G. **Sort the factors by loading size,** makes it easier to think through.

```r
> fa.sort(faresults)  ## where 'faresults' has the results of a factor analysis
> fa.organize(faresults)  ## leaves items in original order
```

H. ..

I. **Kaiser-Meyer-Olkin** test of “sampling adequacy”. Some say don’t extract factors if this is below .50. The higher the better.

```r
> KMO(D2AxS[, 26:31])  ## a few columns of a small data set
```

Kaiser-Meyer-Olkin factor adequacy

Call: KMO(r = D2AxS[, 26:31])

Overall MSA = 0.55

MSA for each item =

<table>
<thead>
<tr>
<th></th>
<th>subnig</th>
<th>vta</th>
<th>acmb</th>
<th>amyg</th>
<th>bs</th>
<th>caud</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.58</td>
<td>0.49</td>
<td>0.57</td>
<td>0.58</td>
<td>0.47</td>
<td>0.56</td>
</tr>
</tbody>
</table>
J. Calculate Cronbach’s alpha (see Revelle’s documentation for other methods that are less entrenched but perhaps better)

First make a matrix with the items in your scale. Then use ‘alpha’.

```r
> library(psych); # just a reminder to you to activate the ‘psych’ package
> fac1=data.frame(R_understandable_language, R_friendly, R_available,
  R_announceresults, R_reportresults, R_sigcontribution_community,
  R_sigcontribution_personal) # put the variables in a data frame

> alphafac1=alpha(fac1,keys=NULL, cumulative=F,na.rm=T)
> alphafac1
```

Reliability analysis

```
Call: alpha(x = fac1, keys = NULL, cumulative = F, na.rm = T)

  raw_alpha std.alpha  G6(smc) average_r S/N   ase  mean  sd
  0.9      0.9    0.92  0.56  8.8 0.034  6.5 1.9

lower alpha upper     95% confidence boundaries
  0.83 0.9 0.96
```

Reliability if an item is dropped:

```
  raw_alpha std.alpha  G6(smc) average_r S/N alpha se
R_understandable_language  0.89  0.89  0.91  0.58  8.4 0.038
R_friendly                0.89  0.89  0.90  0.58  8.2 0.038
R_available               0.88  0.88  0.91  0.56  7.5 0.040
R_announceresults         0.87  0.87  0.89  0.53  6.8 0.041
R_reportresults           0.87  0.87  0.89  0.54  6.9 0.041
R_sigcontribution_community 0.87  0.88  0.90  0.54  7.1 0.041
R_sigcontribution_personal 0.89  0.89  0.91  0.58  8.2 0.039
```

Item statistics

```
 n   r r.cor r.drop mean   sd
R_understandable_language  112 0.72 0.66 0.60 6.7 2.3
R_friendly                113 0.73 0.68 0.61 7.5 2.1
R_available               114 0.79 0.75 0.70 6.4 2.3
R_announceresults         112 0.86 0.86 0.80 6.3 2.4
R_reportresults           115 0.85 0.85 0.79 6.3 2.5
R_sigcontribution_community  114 0.83 0.81 0.78 6.5 2.7
R_sigcontribution_personal  114 0.74 0.68 0.64 5.8 2.9
```

V. Nifty stuff (Not inside ‘psych’ package)

Sometimes we want to split a large sample in order to cross validate a factor solution.

K. Code to split a large enough data set randomly into 2 groups (won’t be exactly equal, but fiddle around until the split is close to equal)

```r
> x=as.matrix(sample(c(0,1),1139, replace=T))
# the data sample has 1139 observations, so create a variable, x, with 1139 randomly sampled 1’s and 0’s.
> mean(x) # find the mean to see how close to an equal split it was
[1] 0.4978051  ## can re-do the split until we get one that is about 50-50
> newdat=cbind(x,asiq); # column bind the new variable with original data
> ncol(newdat) ## original data had 49 columns, checking that now I have 50
[1] 50
> newdat1=subset(newdat, x==1); # now I extract the cases with the 1’s
> nrow(newdat1);
[1] 567  ## there are 567 observations in the data set labeled 1.
> newdat0=subset(newdat, x==0); # extract the cases with the 0’s
> nrow(newdat0);  ## check the number of observations
```
[1] 572  ## there are 572 observations in the data set labeled 0.
> library(MASS)  ## the write.matrix function is in MASS package
> write.matrix(newdat0, file="TA0data.txt", sep=" ")
# save the results for the cases with 0’s Instead of using a blank as the
# separator you can use a comma to create a csv file

> write.matrix(newdat1, file="TA1data.txt", sep=" ")
# save the results for the cases with 1’s.

L. Code to do what
M. more nifty code maybe

1. (blah blah to be continued... perhaps)
2.