PRINCIPAL COMPONENT ANALYSIS (PCA)

CONDUCTING A PRINCIPAL COMPONENT ANALYSIS

- *packages*: GPArotation, magrittr, pacman, psych rio, tidyverse
- *dataset*: import b5.csv
- principal components analysis
    - several different functions available or doing principal component analysis
    - principal component model using default method (`precom`p)
    - get summary statistics for PC
    - screeplot of eigenvalues
- very simple structure (VSS)
    - or use "nfactors" to do the same
- factor analysis
    - calculate and plot factors with `fa()`
- hierarchical clustering
    - hierarchical clustering of items (`iclust()`)
- PC with k factors
    - PCA with no rotation
    - PCA with oblique rotation

- correlations good - for when looking at how one variable here - is connected to one variable there
- sometimes what to look at connections/associations for an entire group of variables
- ex: several questions on a survey - that all assess , more or less, the same thing - but
    - 1) want to be able to average them
    - or2) want to see how they group with one another
- use **principal component analysis** -or- **factor analysis**
    - (2 closely related procedures)

INSTALL AND LOAD PACKAGES

```
pacman::p_load(GPArotation, magrittr, pacman, psych, rio, tidyverse)
```

- **GPArotation** - gradient projection algorithm rotation for FACTORS

LOAD AND PREPARE DATA

Import data from CSV, save as "df":

```
df <- import("data/b5.csv")
```

- result: envir - data: df 18930 obs of 50 variables
- there are **10 variables each - for the 5 personality factors**

Data	
df	18930 obs. of 50 variables

```
> # Get column names
> df %>% colnames()
 [1] "E1"  "E2"  "E3"  "E4"  "E5"  "E6"  "E7"  "E8"  "E9"  "E10" "N1"  "N2"
[13] "N3"  "N4"  "N5"  "N6"  "N7"  "N8"  "N9"  "N10" "A1"  "A2"  "A3"  "A4"
[25] "A5"  "A6"  "A7"  "A8"  "A9"  "A10" "C1"  "C2"  "C3"  "C4"  "C5"  "C6"
[37] "C7"  "C8"  "C9"  "C10" "O1"  "O2"  "O3"  "O4"  "O5"  "O6"  "O7"  "O8"
[49] "O9"  "O10"
```


```
 [1] "E1"  "E2"  "E3"  "E4"  "E5"  "E6"  "E7"  "E8"  "E9"  "E10" "N1"  "N2"
[13] "N3"  "N4"  "N5"  "N6"  "N7"  "N8"  "N9"  "N10" "A1"  "A2"  "A3"  "A4"
[25] "A5"  "A6"  "A7"  "A8"  "A9"  "A10" "C1"  "C2"  "C3"  "C4"  "C5"  "C6"
[37] "C7"  "C8"  "C9"  "C10" "O1"  "O2"  "O3"  "O4"  "O5"  "O6"  "O7"  "O8"
[49] "O9"  "O10"
```

PRINCIPAL COMPONENTS ANALYSIS

- like trying to draw a line thru multidimensional space - that adequately summarises a lot of the variability
- ex HEIGHT WEIGHT - they are associated
    - > can talk about a principal component = that is about SIZE or BIGNESS
    - > use that - instead of these 2 CORRELATED DIMENSIONS
- adv - gives you a little bit less that have to deal with (maybe cancels out some of the noise) but gets to the essential details of your particular analysis

Several dift functions available for doing principal component analysis
Three methods in R:

```
?prcomp       # most common method (in R by default)
?princomp     # slightly dift method - from pre-R language called S (in R by default)
?principal    # method from psych package (favourite)
```

Principal component model using default method:

```
pc <- df %>%
    prcomp(
        center = TRUE       # centers means to 0 (optional)
        scale = TRUE        # sets unit variance (helpful)
    )
```

- **df** - has only the outcome of questions
- **center values** - gives all the same mean of 0
- **scale values** - gives all the same variance and standard deviation of 1
    - important - bc IF your variables are on dift scales THEN the ones that have a larger scale, dominate the analysis (dn want that)
- results: envir - data: **pc large prgroup (5 elements, 8.4 Mb)**

pc	Large prcomp (5 elements, 8.8 MB)

Get summary statistics for pc:

```
summary(pc)
```

- results:

importance of components PC1 PC2 PC3 … PC50
- for each PC shows
  - 1) standard deviation
  - 2) proportion of variance
  - 3) cumulative proportion

```
Importance of components:
                          PC1     PC2     PC3    PC4     PC5    PC6     PC7
Standard deviation     2.837 2.15015 1.94001 1.8842 1.66227 1.2511 1.15342
Proportion of Variance 0.161 0.09246 0.07527 0.0710 0.05526 0.0313 0.02661
Cumulative Proportion  0.161 0.25348 0.32875 0.3998 0.45502 0.4863 0.51293
                           PC8     PC9   PC10    PC11    PC12    PC13    PC14
Standard deviation     1.02448 0.98413 0.9617 0.94714 0.93146 0.91905 0.89562
Proportion of Variance 0.02099 0.01937 0.0185 0.01794 0.01735 0.01689 0.01604
Cumulative Proportion  0.53392 0.55329 0.5718 0.58973 0.60708 0.62398 0.64002
                          PC15   PC16   PC17    PC18    PC19    PC20    PC21
Standard deviation     0.88770 0.8574 0.8544 0.84801 0.82540 0.81318 0.81020
Proportion of Variance 0.01576 0.0147 0.0146 0.01438 0.01363 0.01323 0.01313
Cumulative Proportion  0.65578 0.6705 0.6851 0.69946 0.71309 0.72631 0.73944
                          PC22    PC23   PC24    PC25    PC26    PC27    PC28
Standard deviation     0.79644 0.78210 0.7681 0.76291 0.75388 0.74457 0.72978
Proportion of Variance 0.01269 0.01223 0.0118 0.01164 0.01137 0.01109 0.01065
Cumulative Proportion  0.75213 0.76436 0.7762 0.78780 0.79917 0.81026 0.82091
                          PC29    PC30    PC31    PC32    PC33    PC34    PC35
Standard deviation     0.72285 0.71456 0.70840 0.70125 0.69817 0.69424 0.66920
Proportion of Variance 0.01045 0.01021 0.01004 0.00984 0.00975 0.00964 0.00896
Cumulative Proportion  0.83136 0.84157 0.85161 0.86144 0.87119 0.88083 0.88979
                          PC36    PC37    PC38    PC39    PC40    PC41   PC42
Standard deviation     0.66858 0.65893 0.64839 0.64463 0.63571 0.63006 0.6165
Proportion of Variance 0.00894 0.00868 0.00841 0.00831 0.00808 0.00794 0.0076
Cumulative Proportion  0.89873 0.90741 0.91582 0.92413 0.93221 0.94015 0.9477
                          PC43    PC44    PC45   PC46   PC47    PC48    PC49
Standard deviation     0.61161 0.60308 0.58938 0.5873 0.5702 0.56901 0.55859
Proportion of Variance 0.00748 0.00727 0.00695 0.0069 0.0065 0.00648 0.00624
Cumulative Proportion  0.95523 0.96251 0.96946 0.9764 0.9829 0.98933 0.99557
                          PC50
Standard deviation     0.47050
Proportion of Variance 0.00443
Cumulative Proportion  1.00000
```

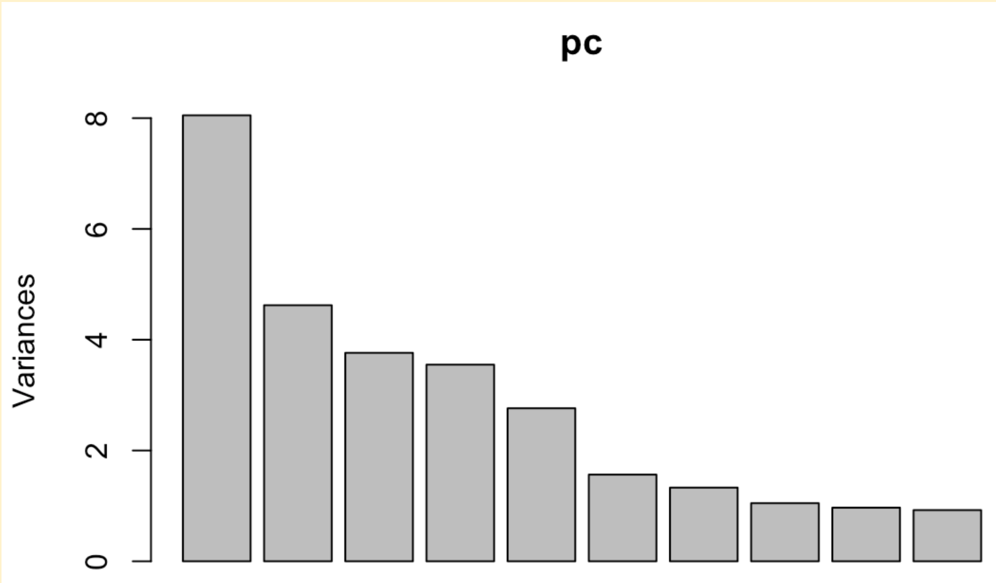
- the principal components - is a way of proportioning the variability that is in the data

Screeplot of eigenvalues:

```
plot(pc)
```

- has to do with the rubble that falls off a side of a cliff. The biggest piles are right next to the cliff, and then they taper down in descending order
- plot has PCs in descending order of proportional relevance
- Result:

-This one is not labelled - but tells you how much variability/variance each component accounts for
-first component - accounts for 8 units of variance (an eigenvalue)
-the next one accounts for 4, … get to 5 very low at end
→ lets us know that - even tho have 50 variables in our data - we might be able to boil it down to a smaller number.



- **scree plot** - is one of the tools - for assessing - how many components you should have in your data
- this **b5 dataset** - is **designed to have 5 components** - so there is a little jump down in relevance, after the 5 PC mark

IF trying to figure out - how many factors/components you should have in your data
THEN there are a few other approaches you can use:

1) **Very Simple Structure (VSS)**
2) **Factor Analysis**
3) **Hierarchical Clustering**
4) **PC with K Factors**

VERY SIMPLE STRUCTURE (VSS)

- use *"very simple structure"* to suggest number of factors
- **MAP** = minimum absolute partial correlation
- **n** is the proposed maximum number of factors
- idea - when do principal component analysis - gives you weights, that can multiply every variable by > to get a new component score for everything
- in practice - people usually put a variable on one component - they average just one score
- ⇒ VSS - is an attempt to find - how many components you need - when you are going to put a variable - only into one component.

```
df %>%
    select(1:50) %>%        # select first 50 variables (all the ones in dataset)
    vss(n = 10)             # run up to 10 possible factors/components
```

- results:

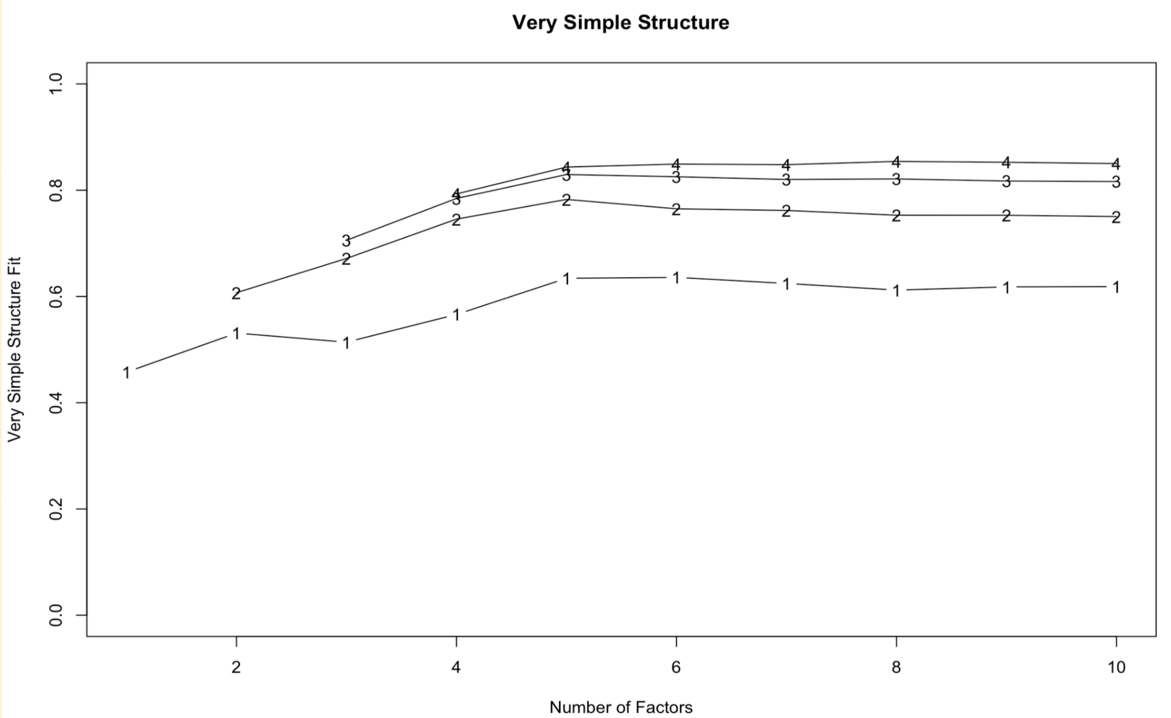
```
Statistics by number of factors
   vss1 vss2    map  dof  chisq prob sqresid  fit RMSEA    BIC  SABIC complex
1  0.46 0.00 0.0234 1175 247942    0      76 0.46 0.105 236370 240104     1.0
2  0.53 0.61 0.0175 1126 181584    0      55 0.61 0.092 170494 174073     1.2
3  0.51 0.67 0.0133 1078 133756    0      41 0.71 0.081 123139 126565     1.6
4  0.57 0.75 0.0103 1031  94542    0      29 0.79 0.069  84388  87664     1.5
5  0.63 0.78 0.0060  985  56723    0      22 0.85 0.055  47022  50152     1.3
6  0.64 0.76 0.0057  940  45066    0      19 0.86 0.050  35809  38796     1.5
7  0.62 0.76 0.0059  896  36826    0      18 0.87 0.046  28002  30849     1.6
8  0.61 0.75 0.0062  853  29070    0      17 0.88 0.042  20670  23380     1.7
9  0.62 0.75 0.0067  811  24933    0      16 0.88 0.040  16946  19523     1.8
10 0.62 0.75 0.0071  770  21043    0      15 0.89 0.037  13460  15907     1.8
   eChisq  SRMR eCRMS   eBIC
1  737929 0.126 0.129 726357
2  454648 0.099 0.103 443558
3  282088 0.078 0.083 271472
4  136940 0.054 0.059 126786
5   50472 0.033 0.037  40771
6   33927 0.027 0.031  24670
7   24042 0.023 0.027  15218
8   18801 0.020 0.024  10400
9   15790 0.018 0.023   7803
10  12722 0.017 0.021   5139
```

graph: Very Simple Structure
x = number of factors
- shows how many factors its adding (from 1 to 10)

y =very simple structure fit
- shows how well the model fits the data - goes from 0 at bottom, which is horrible, to 1 which is perfect (dn expect it to get to 1)

the numbers and lines in the graph - indicate - how many COMPONENTS are we going to allow EACH VARIABLE to contribute
- in practice, it should be just 1 (so ignore the other lines)
- >look at line with 1's - see that at 5 factors, the line flattens out (even goes downhill a bit)
- >> indicates that maybe a 5 factor model is appropriate (which makes sense, bc that is what it was designed for)

```
Very Simple Structure
Call: vss(x = ., n = 10)
Although the VSS complexity 1 shows  6  factors, it is probably more reasonable to think about  2  factors
VSS complexity 2 achieves a maximimum of 0.78  with  5  factors

The Velicer MAP achieves a minimum of 0.01  with  6  factors
BIC achieves a minimum of  13459.82  with  10  factors
Sample Size adjusted BIC achieves a minimum of  15906.84  with  10  factors
```



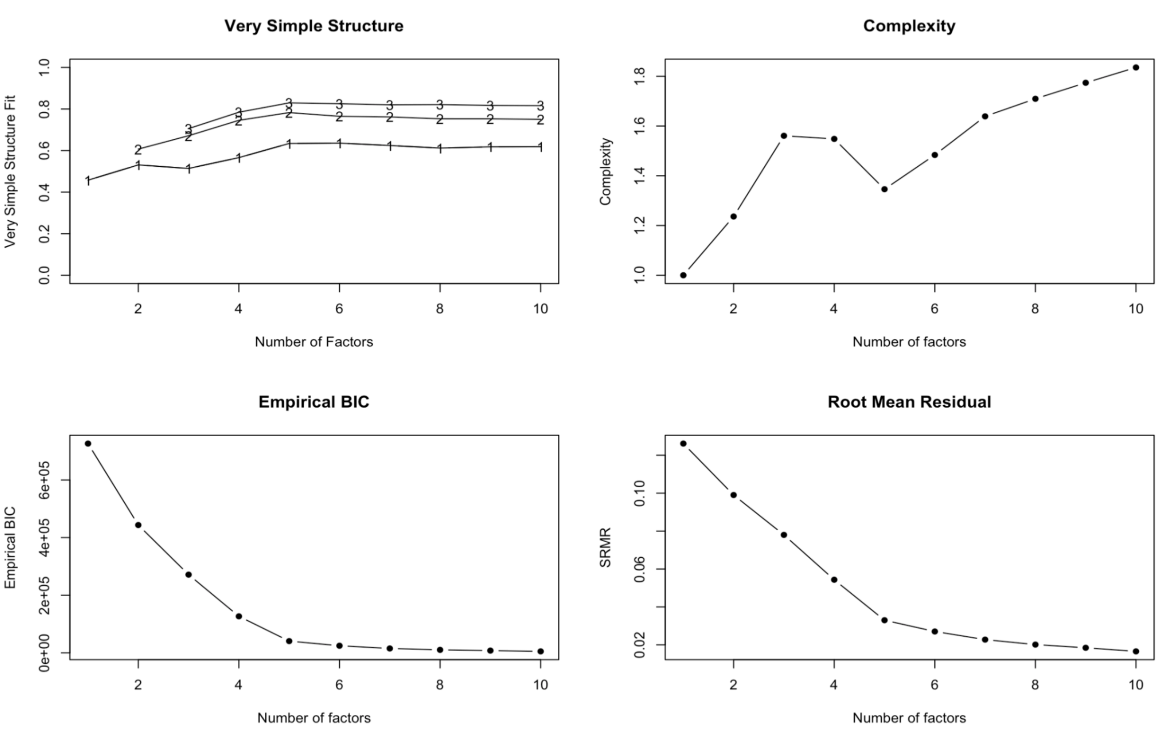
Or use "nfactors" to do the same (includes VSS)

```
df %>%
    select(1:50) %>%
    nfactors(n = 10)
```

- results: 4 graphs

4 graphs

- 1) **very simple structure** (like above) (flatten at 5 factors)
- 2) **complexity** (dips down at 5 factors)
- 3) **empirical BIC** (bayesian information criterion) (flattens out at 5)
- 4) **root mean residual** (bend at 5)
- >> each shows something distinctive going on at 5 factors



- concl: taken together - all would suggest that 5 factors would be an appropriate solution for our data

```
Number of factors
Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
    n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)
VSS complexity 1 achieves a maximimum of Although the vss.max shows  6  factor
s, it is probably more reasonable to think about  2  factors
VSS complexity 2 achieves a maximimum of 0.78  with  5  factors
The Velicer MAP achieves a minimum of 0.01  with  6  factors
Empirical BIC achieves a minimum of  5139.05  with  10  factors
Sample Size adjusted BIC achieves a minimum of  15906.84  with  10  factors
```

FACTOR ANALYSIS

- Factor analysis using **minimum residual** (`minres`) method and **oblimin rotation**, which is useful for simple structure.
- Need to enter desired number of factors (from **VSS** or `nfactors` above)
- closely related to **principal component analysis** (have a dift theory about the relationship bw the factors and the variables) - but often used interchangeably

Calculate and plot factors with `fa()`:

```
df %>%
    select(1:50) %>%              # select variables
    fa(                           # use fa() function
       nfactors = 5,              # use 5 factors
       rotate = "oblimin"         # oblimin oblique rotation
      ) %T>%                # T-pipe
    fa.diagram() %>%              # diagram of factors and variables
    print()                       # print results
```

- **oblimin oblique rotation** (way to simplify the interpretation of the data)
- **T-pipe** (t pipe feeds results to BOTH fa.diagram and print - without stopping in between)
- results:

console: lots of info
- 1) shows 50 variables - along with the 5 factors (50 x 5)
- 2) commonality, uniqueness …

this is not about theory of factor analysis
- is about - how to get the info you need to interpret your results

	MR1	MR2	MR3	MR5	MR4	h2	u2	com
E1	0.69	0.04	-0.03	-0.01	0.01	0.46	0.54	1.0
E2	-0.70	-0.08	-0.04	0.04	0.00	0.48	0.52	1.0
E3	0.63	-0.17	0.16	0.09	-0.06	0.58	0.42	1.3
E4	-0.72	0.05	0.04	0.00	0.03	0.52	0.48	1.0
E5	0.72	0.03	0.12	0.07	0.03	0.60	0.40	1.1
E6	-0.54	0.01	-0.08	0.01	-0.19	0.40	0.60	1.3
E7	0.74	0.00	0.06	0.02	-0.01	0.57	0.43	1.0
E8	-0.60	-0.04	0.11	0.07	-0.01	0.33	0.67	1.1
E9	0.64	0.04	-0.09	-0.02	0.09	0.40	0.60	1.1
E10	-0.65	0.10	0.03	0.00	0.01	0.45	0.55	1.0
N1	-0.06	0.69	0.10	0.05	-0.05	0.49	0.51	1.1
N2	0.07	-0.51	-0.01	-0.09	0.05	0.27	0.73	1.1
N3	-0.11	0.61	0.20	0.10	0.01	0.43	0.57	1.4
N4	0.14	-0.30	-0.07	0.07	-0.07	0.14	0.86	1.8
N5	0.01	0.53	0.01	-0.05	-0.11	0.32	0.68	1.1
N6	0.00	0.75	0.06	-0.01	-0.07	0.57	0.43	1.0
N7	0.06	0.71	-0.05	-0.08	0.02	0.52	0.48	1.1
N8	0.05	0.74	-0.06	-0.08	0.01	0.58	0.42	1.0
N9	0.04	0.73	-0.15	0.04	0.00	0.54	0.46	1.1
N10	-0.21	0.58	0.03	-0.11	0.08	0.47	0.53	1.4
A1	0.05	0.09	-0.44	0.02	-0.07	0.20	0.80	1.2
A2	0.28	-0.04	0.50	-0.04	0.05	0.41	0.59	1.6
A3	0.17	0.27	-0.41	-0.15	0.09	0.27	0.73	2.6
A4	-0.06	0.03	0.80	-0.01	-0.01	0.62	0.38	1.0
A5	-0.06	0.04	-0.66	0.05	0.00	0.45	0.55	1.0
A6	-0.05	0.12	0.61	0.02	-0.08	0.37	0.63	1.1
A7	-0.23	0.09	-0.61	0.05	-0.01	0.51	0.49	1.4
A8	0.05	-0.02	0.58	0.05	0.02	0.36	0.64	1.0
A9	0.04	0.10	0.70	0.03	0.04	0.51	0.49	1.1
A10	0.28	-0.08	0.34	0.11	0.06	0.31	0.69	2.4
C1	0.01	-0.02	-0.04	0.60	0.12	0.39	0.61	1.1
C2	0.05	0.04	0.08	-0.54	0.12	0.31	0.69	1.2
C3	-0.08	0.05	0.07	0.40	0.27	0.24	0.76	2.0
C4	-0.03	0.30	0.01	-0.53	0.02	0.45	0.55	1.6
C5	0.08	0.01	0.00	0.63	-0.08	0.41	0.59	1.1
C6	0.02	0.10	0.05	-0.59	0.06	0.38	0.62	1.1
C7	-0.06	0.14	0.00	0.56	0.05	0.30	0.70	1.2
C8	-0.02	0.16	-0.11	-0.45	-0.03	0.30	0.70	1.4
C9	0.05	0.11	0.04	0.64	-0.03	0.41	0.59	1.1
C10	0.00	0.05	0.03	0.47	0.24	0.28	0.72	1.5
O1	-0.03	-0.03	-0.04	0.02	0.60	0.36	0.64	1.0
O2	0.06	0.21	-0.02	0.05	-0.56	0.36	0.64	1.3
O3	-0.02	0.10	0.07	-0.10	0.53	0.30	0.70	1.2
O4	0.09	0.13	-0.13	0.11	-0.47	0.26	0.74	1.5
O5	0.15	-0.01	-0.06	0.14	0.58	0.41	0.59	1.3
O6	-0.04	0.04	-0.08	0.06	-0.49	0.27	0.73	1.1
O7	0.02	-0.10	-0.03	0.17	0.49	0.30	0.70	1.3
O8	-0.04	0.08	-0.11	-0.05	0.56	0.32	0.68	1.1
O9	-0.19	0.15	0.21	0.04	0.35	0.20	0.80	2.7
O10	0.13	0.02	0.00	0.02	0.66	0.48	0.52	1.1

```
                       MR1  MR2  MR3  MR5  MR4
SS loadings           5.07 4.55 3.74 3.27 3.20
Proportion Var        0.10 0.09 0.07 0.07 0.06
Cumulative Var        0.10 0.19 0.27 0.33 0.40
Proportion Explained  0.26 0.23 0.19 0.16 0.16
Cumulative Proportion 0.26 0.49 0.67 0.84 1.00
```

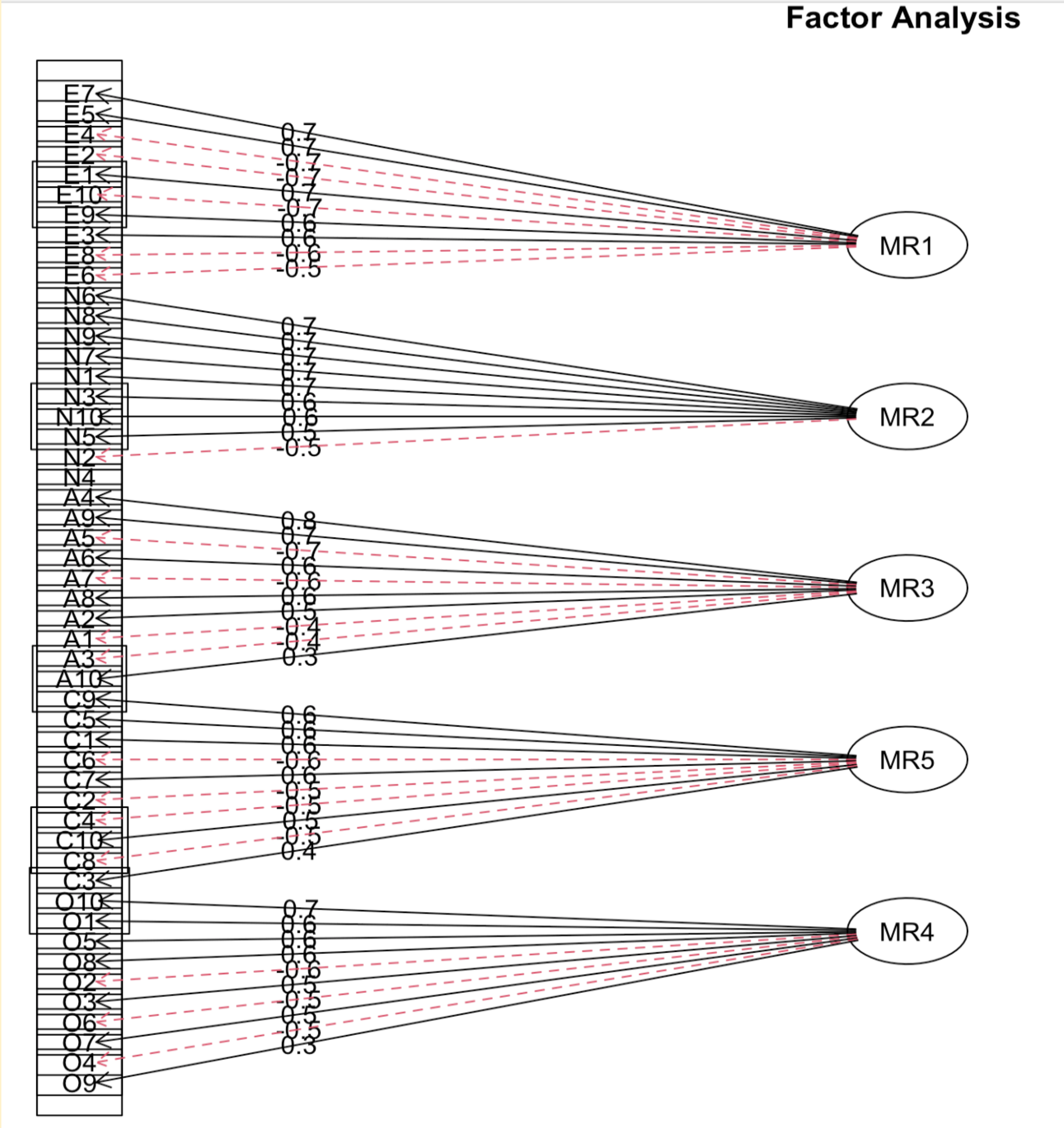
```
 With factor correlations of 
      MR1   MR2   MR3   MR5   MR4
MR1  1.00 -0.24  0.25  0.09  0.16
MR2 -0.24  1.00 -0.03 -0.24 -0.08
MR3  0.25 -0.03  1.00  0.14  0.07
MR5  0.09 -0.24  0.14  1.00  0.05
MR4  0.16 -0.08  0.07  0.05  1.00
```

```
Mean item complexity =  1.3
Test of the hypothesis that 5 factors are sufficient.

df null model =  1225  with the objective function =  19.15 with Chi Square =  362087.3
df of  the model are 985  and the objective function was  3 

The root mean square of the residuals (RMSR) is  0.03 
The df corrected root mean square of the residuals is  0.04 

The harmonic n.obs is  18930 with the empirical chi square  50472.27  with prob <  0 
The total n.obs was  18930  with Likelihood Chi Square =  56722.95  with prob <  0 

Tucker Lewis Index of factoring reliability =  0.808
RMSEA index =  0.055  and the 90 % confidence intervals are  0.054 0.055
BIC =  47022.17
Fit based upon off diagonal values = 0.97
Measures of factor score adequacy             
```

```
                                                 MR1  MR2  MR3  MR5
Correlation of (regression) scores with factors  0.95 0.95 0.94 0.91
Multiple R square of scores with factors         0.91 0.90 0.87 0.84
Minimum correlation of possible factor scores    0.81 0.79 0.75 0.67
                                                 MR4
Correlation of (regression) scores with factors  0.91
Multiple R square of scores with factors         0.83
Minimum correlation of possible factor scores    0.66
```

<u>chart</u>: shows 5 dift factors MR1 .. MR5 on right - individual variables on left

- **black line** = positive connections
- dotted red = negative connections (but potentially still meaningful associations



- <u>concl:</u> this falls into the categories we expected - very easily

HIERARCHICAL CLUSTERING

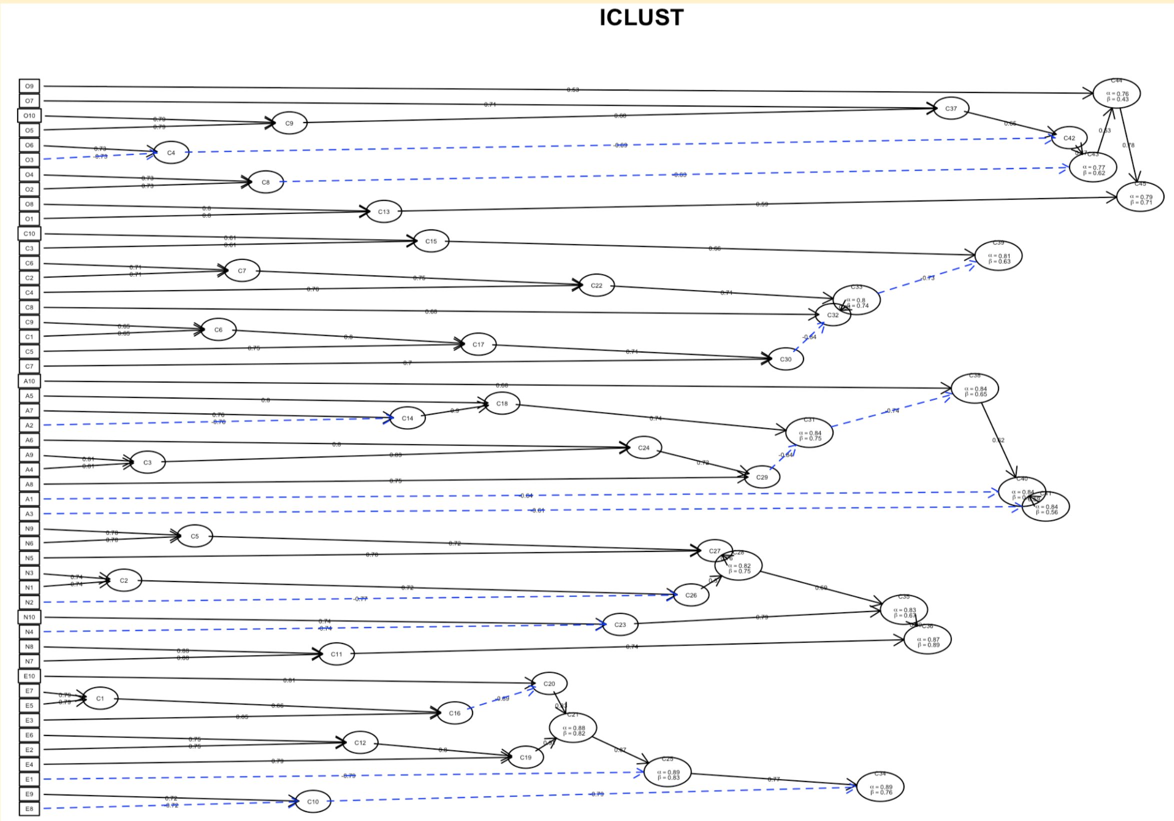
Hierarchical clustering of items with `iclust()`

```
df %>%
    select(1:50) %>%
    iclust()
```

- so far: done this before - with the clustering of CASES (groups of people , or- units of observation)
- now: do with VARIABLES (dift approach)
- call `iclust` function -
- results:

-console: lots output

-graph: 50 dift variables on left > at each step (how they get combined with one another)



- concl: fall into naturally relevant clusters - for how the 50 questions go into the 5 PERSONALITY FACTORS

PC WITH K FACTORS

Principal components with set number , K, factors

1)First, PCA with NO rotation , specify 5 factors:

```
df %>% principal(nfactors = 5)   # from psych package
```

- results: console

similar to other outputs (50 variables - 5 factors)

```
Principal Components Analysis
Call: principal(r = ., nfactors = 5)
Standardized loadings (pattern matrix) based upon correlation matrix
```

```
      RC1   RC2   RC5   RC3   RC4   h2   u2 com
E1   0.71 -0.05  0.05  0.01  0.03 0.52 0.48 1.0
E2  -0.72  0.00 -0.12  0.03 -0.03 0.53 0.47 1.1
E3   0.67 -0.26  0.26  0.13 -0.02 0.60 0.40 1.7
E4  -0.74  0.15 -0.05 -0.02  0.00 0.57 0.43 1.1
E5   0.74 -0.08  0.22  0.10  0.07 0.62 0.38 1.3
E6  -0.60  0.08 -0.16 -0.02 -0.23 0.45 0.55 1.5
E7   0.75 -0.10  0.16  0.05  0.03 0.61 0.39 1.1
E8  -0.62  0.02  0.06  0.07 -0.03 0.40 0.60 1.1
E9   0.67 -0.04 -0.03 -0.01  0.12 0.46 0.54 1.1
E10 -0.68  0.19 -0.06 -0.02 -0.02 0.50 0.50 1.2
N1  -0.11  0.73  0.07 -0.01 -0.07 0.55 0.45 1.1
N2   0.11 -0.56  0.01 -0.05  0.07 0.33 0.67 1.1
N3  -0.14  0.66  0.18  0.06 -0.01 0.49 0.51 1.3
N4   0.16 -0.37 -0.04  0.10 -0.07 0.18 0.82 1.7
N5  -0.04  0.59 -0.02 -0.11 -0.14 0.38 0.62 1.2
N6  -0.06  0.77  0.03 -0.08 -0.09 0.61 0.39 1.1
N7   0.00  0.73 -0.09 -0.15  0.00 0.57 0.43 1.1
N8  -0.01  0.76 -0.09 -0.16 -0.02 0.61 0.39 1.1
N9  -0.04  0.74 -0.19 -0.04 -0.03 0.58 0.42 1.1
N10 -0.26  0.65 -0.04 -0.17  0.06 0.52 0.48 1.5
A1   0.00  0.08 -0.50 -0.01 -0.09 0.26 0.74 1.1
A2   0.35 -0.05  0.57  0.00  0.09 0.46 0.54 1.7
A3   0.13  0.27 -0.46 -0.20  0.09 0.35 0.65 2.3
A4   0.04  0.06  0.80  0.04  0.02 0.65 0.35 1.0
A5  -0.13  0.02 -0.71  0.01 -0.02 0.52 0.48 1.1
A6  -0.01  0.16  0.65  0.03 -0.08 0.45 0.55 1.2
A7  -0.31  0.10 -0.67  0.00 -0.05 0.56 0.44 1.5
A8   0.12 -0.02  0.64  0.09  0.04 0.43 0.57 1.1
A9   0.12  0.12  0.73  0.07  0.07 0.56 0.44 1.1
A10  0.36 -0.13  0.42  0.15  0.09 0.35 0.65 2.6
C1   0.05 -0.10  0.01  0.65  0.12 0.45 0.55 1.1
C2   0.06  0.10  0.05 -0.59  0.14 0.38 0.62 1.2
C3  -0.04  0.02  0.09  0.47  0.29 0.31 0.69 1.8
C4  -0.06  0.38 -0.04 -0.59  0.02 0.49 0.51 1.8
C5   0.09 -0.08  0.06  0.67 -0.10 0.48 0.52 1.1
C6   0.00  0.17  0.01 -0.64  0.07 0.44 0.56 1.2
C7  -0.05  0.09  0.03  0.61  0.04 0.38 0.62 1.1
C8  -0.06  0.23 -0.17 -0.52 -0.03 0.36 0.64 1.7
C9   0.07  0.04  0.09  0.68 -0.04 0.48 0.52 1.1
C10  0.04  0.00  0.06  0.53  0.25 0.35 0.65 1.5
O1   0.03 -0.04 -0.04  0.05  0.65 0.43 0.57 1.0
O2   0.00  0.23 -0.03  0.01 -0.61 0.43 0.57 1.3
O3   0.04  0.12  0.07 -0.09  0.59 0.37 0.63 1.2
O4   0.03  0.13 -0.13  0.08 -0.54 0.34 0.66 1.3
O5   0.22 -0.06 -0.03  0.18  0.62 0.47 0.53 1.5
O6  -0.10  0.05 -0.09  0.04 -0.56 0.33 0.67 1.1
O7   0.08 -0.14 -0.01  0.22  0.54 0.36 0.64 1.5
O8   0.00  0.09 -0.13 -0.05  0.61 0.40 0.60 1.1
O9  -0.17  0.19  0.21  0.05  0.40 0.27 0.73 2.5
O10  0.20 -0.02  0.02  0.05  0.70 0.53 0.47 1.2
```

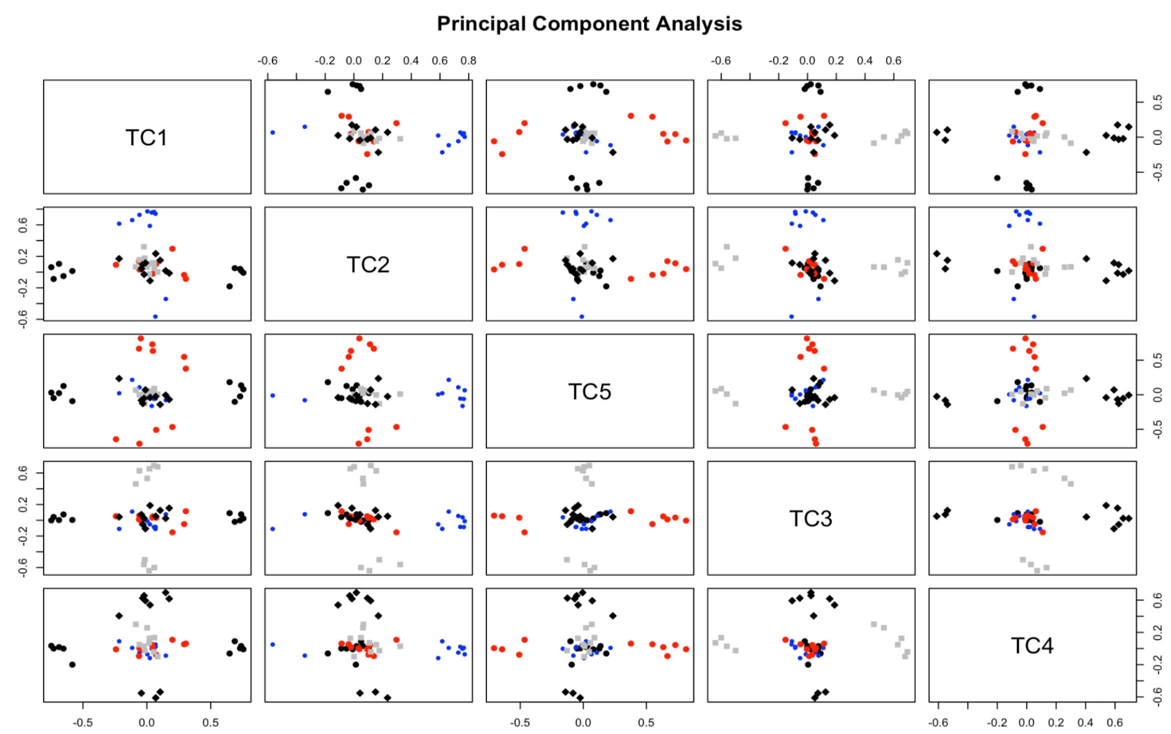
```
                       RC1  RC2  RC5  RC3  RC4
SS loadings           5.52 5.15 4.35 3.91 3.82
Proportion Var        0.11 0.10 0.09 0.08 0.08
Cumulative Var        0.11 0.21 0.30 0.38 0.46
Proportion Explained  0.24 0.23 0.19 0.17 0.17
Cumulative Proportion 0.24 0.47 0.66 0.83 1.00

Mean item complexity =  1.3
Test of the hypothesis that 5 components are sufficient.

The root mean square of the residuals (RMSR) is  0.04
 with the empirical chi square  81657.81  with prob <  0

Fit based upon off diagonal values = 0.95
```

2)second, PCA with oblimin (oblique) rotation:

```
df %>%
    principal(
        nfactors = 5,
        rotate = "oblimin"
    ) %>%
    plot()                          # plot position of variables on components
```

- result:

plot - 5 components on a diagonal
- plots show how dift variables - load on the dift factors



SUMMARY:
- another way of looking at the associations of individual variables, and how they might be combined - into larger factors or components, which
    - 1) reduces the number of the number of things we have to deal with
    - and 2) can help cancel out some of the idiosyncratic variation of individual variables - to get a clearer look on the signal on the noise in your data