Principal Components I

Mike Strube

September 17, 2018

1 Preliminaries

In this section, the RStudio workspace and console panes are cleared of old output, variables, and other miscellaneous debris. Packages are loaded.

```
options(replace.assign = TRUE, width = 65, digits = 4, scipen = 4, fig.width = 4,
    fig.height = 4)
# Clear the workspace and console.
rm(list = ls(all = TRUE))
cat("\f")
```

```
# Turn off showing of significance asterisks.
options(show.signif.stars = F)
# Set the contrast option; important for ANOVAs.
options(contrasts = c("contr.sum", "contr.poly"))
how_long <- Sys.time()
set.seed(123)
library(knitr)
```

1.1 Packages

```
library(psych)
library(ggplot2)

##
## Attaching package:  ’ggplot2’
## The following objects are masked from ’package:psych’:
##
##     %+%, alpha

library(factoextra)

## Warning:  package ’factoextra’ was built under R version 3.5.1
## Welcome!  Related Books:  ‘Practical Guide To Cluster Analysis in R‘ at https://goo.gl/13EFCZ

library(FactoMineR)

## Warning:  package ’FactoMineR’ was built under R version 3.5.1

library(reshape2)
library(GGally)
library(MASS)
library(parallel)
library(MVN)
```

```
## sROC 0.1-2 loaded

library(qqplotr)
library(arm)

## Loading required package:  Matrix
## Loading required package:  lme4
##
## arm (Version 1.10-1, built:  2018-4-12)
## Working directory is C:/Courses/Psychology 516/PowerPoint/2018
##
## Attaching package:  'arm'
## The following objects are masked from 'package:psych':
##
##    logit, rescale, sim

library(psych)
library(lme4)
library(lmtest)

## Loading required package:  zoo
##
## Attaching package:  'zoo'
## The following objects are masked from 'package:base':
##
##    as.Date, as.Date.numeric

library(car)

## Loading required package:  carData
##
## Attaching package:  'car'
## The following object is masked from 'package:arm':
##
##    logit
## The following object is masked from 'package:psych':
##
##    logit

library(emmeans)

## Warning:  package 'emmeans' was built under R version 3.5.1
## NOTE: As of emmeans versions > 1.2.3,
##       The 'cld' function will be deprecated in favor of 'CLD'.
##       You may use 'cld' only if you have package:multcomp attached.
##
## Attaching package:  'emmeans'
## The following object is masked from 'package:GGally':
##
##    pigs

library(multcomp)

## Loading required package:  mvtnorm
## Loading required package:  survival
## Loading required package:  TH.data
##
## Attaching package:  'TH.data'
## The following object is masked from ’package:MASS’:
##
##     geyser
##
## Attaching package:  ’multcomp’
## The following object is masked from ’package:emmeans’:
##
##     cld

library(lm.beta)
library(pls)

##
## Attaching package:  ’pls’
## The following objects are masked from ’package:arm’:
##
##     coefplot, corrplot
## The following object is masked from ’package:stats’:
##
##     loadings

library(polycor)

## Warning:  package ’polycor’ was built under R version 3.5.1
##
## Attaching package:  ’polycor’
## The following object is masked from ’package:psych’:
##
##     polyserial
```

2 Simplified Composites

2.1 Data Generation

To explore forming composites of different types, we'll use data similar to what we used to examine outlier detection, with some modification. We will generate a random sample of 500 cases for 9 standard normal variables from a population having moderate correlations (.45 to .70) among items in partially overlapping sets (variables 1-4, variables 4-7, variables, 7-9). So that the matrix is positive definite, we need to allow some correlation among variables across sets (.10 in this case). A 10th variable is included to serve as an outcome variable to explore the impact of different composites on regression estimates.

$$R = \begin{bmatrix} 1.00 & 0.50 & 0.45 & 0.45 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.55 \\ 0.50 & 1.00 & 0.60 & 0.60 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.50 \\ 0.45 & 0.60 & 1.00 & 0.55 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.55 \\ 0.45 & 0.60 & 0.55 & 1.00 & 0.50 & 0.60 & 0.50 & 0.00 & 0.00 & 0.55 \\ 0.00 & 0.00 & 0.00 & 0.50 & 1.00 & 0.50 & 0.65 & 0.50 & 0.00 & 0.45 \\ 0.00 & 0.00 & 0.00 & 0.60 & 0.50 & 1.00 & 0.50 & 0.00 & 0.00 & 0.50 \\ 0.00 & 0.00 & 0.00 & 0.50 & 0.65 & 0.50 & 1.00 & 0.50 & 0.50 & 0.65 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.50 & 1.00 & 0.70 & 0.55 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.50 & 0.70 & 1.00 & 0.60 \\ 0.55 & 0.50 & 0.55 & 0.55 & 0.45 & 0.50 & 0.65 & 0.55 & 0.60 & 1.00 \end{bmatrix}$$

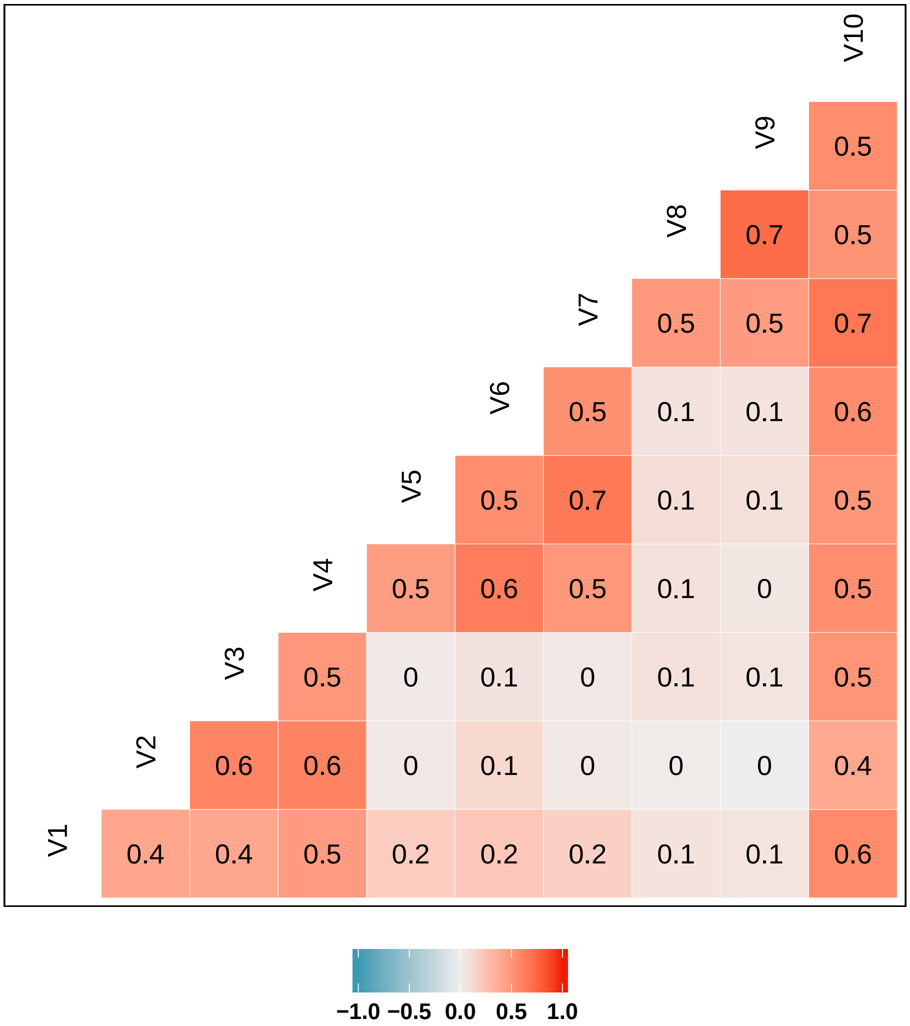
```
means <- matrix(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0))
sigma <- matrix(c(1, 0.5, 0.45, 0.45, 0.1, 0.1, 0.1, 0.1, 0.1, 0.55,
    0.5, 1, 0.6, 0.6, 0.1, 0.1, 0.1, 0.1, 0.1, 0.5, 0.45, 0.6, 1,
    0.55, 0.1, 0.1, 0.1, 0.1, 0.1, 0.55, 0.45, 0.6, 0.55, 1, 0.5,
    0.6, 0.5, 0.1, 0.1, 0.55, 0.1, 0.1, 0.1, 0.5, 1, 0.5, 0.65, 0.1,
    0.1, 0.45, 0.1, 0.1, 0.1, 0.6, 0.5, 1, 0.5, 0.1, 0.1, 0.5, 0.1,
    0.1, 0.1, 0.5, 0.65, 0.5, 1, 0.5, 0.5, 0.65, 0.1, 0.1, 0.1, 0.1,
    0.1, 0.1, 0.5, 1, 0.7, 0.55, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.5,
    0.7, 1, 0.6, 0.55, 0.5, 0.55, 0.55, 0.45, 0.5, 0.65, 0.55, 0.6,
    1), nrow = 10, ncol = 10, byrow = TRUE)
Data <- mvrnorm(500, means, sigma)
Data <- as.data.frame(Data)
```

2.2 Correlations

A heat map for the correlation matrix easily identifies the pattern of correlations in the simulated data, including the relations of the first nine variables with the last one.

```
ggcorr(Data, label = TRUE, angle = 90, hjust = 0.1, size = 4, digits = 2) +
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
        size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
        0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
        legend.title = element_blank()) + ggtitle("Intercorrelations Among Measures")
```

Intercorrelations Among Measures



	V2	V3	V4	V5	V6	V7	V8	V9	V10
V9									0.5
V8								0.7	0.5
V7							0.5	0.5	0.7
V6						0.5	0.1	0.1	0.6
V5					0.5	0.7	0.1	0.1	0.5
V4				0.5	0.6	0.5	0.1	0	0.5
V3			0.5	0	0.1	0	0.1	0.1	0.5
V2		0.6	0.6	0	0.1	0	0	0	0.4
V1	0.4	0.4	0.5	0.2	0.2	0.2	0.1	0.1	0.6

2.3 Forming Composites

Sometimes researchers will use principal components analysis to determine how composite scores should be created, but then will create these composites as simple sums of the original variables rather than optimally weighted principal component scores. Three non-optimal rules were used to derive composites:

Use all items but add or subtract depending on the sign of the loading on a component.
Use only those items that load at least .30 in absolute value.
Use only those items that load at least .50 in absolute value.

These composites will be compared to principal component scores.

2.3.1 Extract Three Principal Components

```
PCA <- principal(Data[, c(1:9)], nfactors = 3, rotate = "none", residuals = TRUE,
    scores = TRUE)
PCA

## Principal Components Analysis
## Call: principal(r = Data[, c(1:9)], nfactors = 3, residuals = TRUE,
##     rotate = "none", scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##     PC1   PC2   PC3   h2   u2 com
## V1 0.56 -0.38  0.16 0.49 0.51 2.0
## V2 0.51 -0.61  0.33 0.75 0.25 2.5
## V3 0.48 -0.55  0.43 0.72 0.28 2.9
## V4 0.86 -0.32 -0.13 0.86 0.14 1.3
## V5 0.65  0.24 -0.52 0.74 0.26 2.2
## V6 0.69  0.09 -0.50 0.73 0.27 1.9
## V7 0.75  0.52 -0.14 0.85 0.15 1.9
## V8 0.39  0.61  0.57 0.85 0.15 2.7
## V9 0.37  0.63  0.56 0.84 0.16 2.6
##
##                        PC1  PC2  PC3
## SS loadings           3.28 2.02 1.52
## Proportion Var        0.36 0.22 0.17
## Cumulative Var        0.36 0.59 0.76
## Proportion Explained  0.48 0.30 0.22
## Cumulative Proportion 0.48 0.78 1.00
##
## Mean item complexity =  2.2
## Test of the hypothesis that 3 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  160.1  with prob <  5.1e-28
##
## Fit based upon off diagonal values = 0.96
```

Create the composites and add them to the data frame.

```
PC <- cbind(Data, PCA$scores)
PC$Unit_1 <- PC$V1 + PC$V2 + PC$V3 + PC$V4 + PC$V5 + PC$V6 + PC$V7 +
    PC$V8 + PC$V9
PC$Unit_2 <- -PC$V1 - PC$V2 - PC$V3 - PC$V4 + PC$V5 + PC$V6 + PC$V7 +
    PC$V8 + PC$V9
PC$Unit_3 <- PC$V1 + PC$V2 + PC$V3 - PC$V4 - PC$V5 - PC$V6 - PC$V7 +
    PC$V8 + PC$V9
PC$L30_1 <- PC$V1 + PC$V2 + PC$V3 + PC$V4 + PC$V5 + PC$V6 + PC$V7 +
    PC$V8 + PC$V9
PC$L30_2 <- -PC$V1 - PC$V2 - PC$V3 - PC$V4 + PC$V7 + PC$V8 + PC$V9
PC$L30_3 <- PC$V2 + PC$V3 - PC$V5 - PC$V6 + PC$V8 + PC$V9
PC$L50_1 <- PC$V1 + PC$V2 + PC$V4 + PC$V5 + PC$V6 + PC$V7
PC$L50_2 <- -PC$V2 - PC$V3 + PC$V7 + PC$V8 + PC$V9
PC$L50_3 <- -PC$V5 - PC$V6 + PC$V8 + PC$V9
```

2.3.2 Descriptive Statistics

```
describe(PC[, c(11:22)])

##        vars   n  mean   sd median trimmed  mad    min   max
## PC1       1 500  0.00 1.00   0.00    0.01 0.94  -3.17  2.64
## PC2       2 500  0.00 1.00   0.00   -0.01 0.99  -2.71  2.70
## PC3       3 500  0.00 1.00   0.04    0.00 0.99  -2.67  3.12
## Unit_1    4 500 -0.20 5.26  -0.04   -0.16 5.12 -17.14 14.35
## Unit_2    5 500 -0.05 4.04   0.16   -0.05 4.11 -12.89 10.84
## Unit_3    6 500  0.09 3.54   0.23    0.09 3.46  -9.70 10.59
## L30_1     7 500 -0.20 5.26  -0.04   -0.16 5.12 -17.14 14.35
## L30_2     8 500  0.03 3.69   0.04    0.00 3.58  -8.88 11.16
## L30_3     9 500  0.00 2.96   0.16   -0.01 2.93  -7.48  9.23
## L50_1    10 500 -0.15 4.07  -0.18   -0.14 3.82 -12.65 10.81
## L50_2    11 500  0.00 3.03   0.03   -0.02 2.95  -7.87  7.94
## L50_3    12 500  0.04 2.43   0.17    0.03 2.44  -7.60  8.37
##        range  skew kurtosis   se
## PC1     5.81 -0.04     0.01 0.04
## PC2     5.41  0.07    -0.09 0.04
## PC3     5.79  0.03     0.02 0.04
## Unit_1 31.49 -0.07    -0.05 0.24
## Unit_2 23.73 -0.04    -0.12 0.18
## Unit_3 20.29  0.01     0.02 0.16
## L30_1  31.49 -0.07    -0.05 0.24
## L30_2  20.03  0.14    -0.02 0.17
## L30_3  16.71  0.04    -0.06 0.13
## L50_1  23.46 -0.01     0.06 0.18
## L50_2  15.81  0.06    -0.14 0.14
## L50_3  15.97  0.09     0.15 0.11
```

2.3.3 Correlations

```
round(cor(PC[, c(11:13)]), 3)

##     PC1 PC2 PC3
## PC1   1   0   0
## PC2   0   1   0
## PC3   0   0   1

round(cor(PC[, c(14:16)]), 3)

##        Unit_1 Unit_2 Unit_3
## Unit_1  1.000  0.157 -0.057
## Unit_2  0.157  1.000 -0.421
## Unit_3 -0.057 -0.421  1.000

round(cor(PC[, c(17:19)]), 3)

##        L30_1  L30_2  L30_3
## L30_1  1.000 -0.150  0.275
## L30_2 -0.150  1.000 -0.033
## L30_3  0.275 -0.033  1.000
```

```
round(cor(PC[, c(20:22)]), 3)

##        L50_1 L50_2  L50_3
## L50_1  1.000 0.080 -0.431
## L50_2  0.080 1.000  0.415
## L50_3 -0.431 0.415  1.000

round(cor(PC[, c(11:22)]), 3)

##           PC1    PC2    PC3 Unit_1 Unit_2 Unit_3  L30_1  L30_2
## PC1     1.000  0.000  0.000  0.987  0.129 -0.187  0.987 -0.216
## PC2     0.000  1.000  0.000  0.057  0.967 -0.226  0.057  0.972
## PC3     0.000  0.000  1.000  0.145 -0.194  0.940  0.145  0.061
## Unit_1  0.987  0.057  0.145  1.000  0.157 -0.057  1.000 -0.150
## Unit_2  0.129  0.967 -0.194  0.157  1.000 -0.421  0.157  0.903
## Unit_3 -0.187 -0.226  0.940 -0.057 -0.421  1.000 -0.057 -0.128
## L30_1   0.987  0.057  0.145  1.000  0.157 -0.057  1.000 -0.150
## L30_2  -0.216  0.972  0.061 -0.150  0.903 -0.128 -0.150  1.000
## L30_3   0.141 -0.071  0.978  0.275 -0.237  0.888  0.275 -0.033
## L50_1   0.968 -0.097 -0.201  0.921  0.067 -0.341  0.921 -0.315
## L50_2   0.186  0.964  0.083  0.247  0.926 -0.169  0.247  0.899
## L50_3  -0.228  0.381  0.886 -0.080  0.156  0.779 -0.080  0.474
##         L30_3  L50_1  L50_2  L50_3
## PC1     0.141  0.968  0.186 -0.228
## PC2    -0.071 -0.097  0.964  0.381
## PC3     0.978 -0.201  0.083  0.886
## Unit_1  0.275  0.921  0.247 -0.080
## Unit_2 -0.237  0.067  0.926  0.156
## Unit_3  0.888 -0.341 -0.169  0.779
## L30_1   0.275  0.921  0.247 -0.080
## L30_2  -0.033 -0.315  0.899  0.474
## L30_3   1.000 -0.057  0.026  0.808
## L50_1  -0.057  1.000  0.080 -0.431
## L50_2   0.026  0.080  1.000  0.415
## L50_3   0.808 -0.431  0.415  1.000
```

Only the PC scores are independent; the other composites are moderately correlated and not always in a consistent direction. Smaller components are not preserved as well in the simpler composites.

2.4 Multiple Regressions with Composite Scores

Here the composites are used to predict an outcome variable (Variable 10). The relative strength of the predictors is compared to indicate the potential consequences of using sub-optimal composites.

```
LM_1 <- lm(PC$V10 ~ PC1 + PC2 + PC3, data = PC)
summary(LM_1)

##
## Call:
## lm(formula = PC$V10 ~ PC1 + PC2 + PC3, data = PC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4149 -0.2630  0.0098  0.2896  1.2208
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.0154     0.0188   -0.82     0.41
## PC1           0.8427     0.0188   44.73  < 2e-16
## PC2           0.1229     0.0188    6.53  1.7e-10
## PC3           0.2213     0.0188   11.74  < 2e-16
##
## Residual standard error: 0.421 on 496 degrees of freedom
## Multiple R-squared:  0.815,Adjusted R-squared:  0.814
## F-statistic:  727 on 3 and 496 DF,  p-value: <2e-16
```

```
summary(lm.beta(LM_1))
```

```
##
## Call:
## lm(formula = PC$V10 ~ PC1 + PC2 + PC3, data = PC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4149 -0.2630  0.0098  0.2896  1.2208
##
## Coefficients:
##             Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept)  -0.0154       0.0000     0.0188   -0.82     0.41
## PC1           0.8427       0.8645     0.0188   44.73  < 2e-16
## PC2           0.1229       0.1261     0.0188    6.53  1.7e-10
## PC3           0.2213       0.2270     0.0188   11.74  < 2e-16
##
## Residual standard error: 0.421 on 496 degrees of freedom
## Multiple R-squared:  0.815,Adjusted R-squared:  0.814
## F-statistic:  727 on 3 and 496 DF,  p-value: <2e-16
```

```
LM_2 <- lm(PC$V10 ~ Unit_1 + Unit_2 + Unit_3, data = PC)
summary(LM_2)
```

```
##
## Call:
## lm(formula = PC$V10 ~ Unit_1 + Unit_2 + Unit_3, data = PC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2968 -0.2387 -0.0089  0.2587  1.2009
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01538    0.01690    0.91     0.36
## Unit_1       0.16546    0.00326   50.82  < 2e-16
## Unit_2       0.03568    0.00466    7.66  9.8e-14
## Unit_3       0.04978    0.00527    9.45  < 2e-16
##
## Residual standard error: 0.377 on 496 degrees of freedom
## Multiple R-squared:  0.851,Adjusted R-squared:  0.85
## F-statistic:  944 on 3 and 496 DF,  p-value: <2e-16
```

```
summary(lm.beta(LM_2))

##
## Call:
## lm(formula = PC$V10 ~ Unit_1 + Unit_2 + Unit_3, data = PC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2968 -0.2387 -0.0089  0.2587  1.2009
##
## Coefficients:
##             Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept)  0.01538      0.00000    0.01690    0.91     0.36
## Unit_1       0.16546      0.89207    0.00326   50.82  < 2e-16
## Unit_2       0.03568      0.14802    0.00466    7.66  9.8e-14
## Unit_3       0.04978      0.18061    0.00527    9.45  < 2e-16
##
## Residual standard error: 0.377 on 496 degrees of freedom
## Multiple R-squared:  0.851,Adjusted R-squared:  0.85
## F-statistic:  944 on 3 and 496 DF,  p-value: <2e-16

LM_3 <- lm(PC$V10 ~ L30_1 + L30_2 + L30_3, data = PC)
summary(LM_3)

##
## Call:
## lm(formula = PC$V10 ~ L30_1 + L30_2 + L30_3, data = PC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3445 -0.2466  0.0085  0.2674  1.0936
##
## Coefficients:
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  0.01775    0.01801    0.99   0.32484
## L30_1        0.16678    0.00360   46.29   < 2e-16
## L30_2        0.02266    0.00493    4.59 0.0000056
## L30_3        0.02242    0.00632    3.55   0.00043
##
## Residual standard error: 0.402 on 496 degrees of freedom
## Multiple R-squared:  0.831,Adjusted R-squared:  0.83
## F-statistic:  811 on 3 and 496 DF,  p-value: <2e-16

summary(lm.beta(LM_3))

##
## Call:
## lm(formula = PC$V10 ~ L30_1 + L30_2 + L30_3, data = PC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3445 -0.2466  0.0085  0.2674  1.0936
##
## Coefficients:
##             Estimate Standardized Std. Error t value  Pr(>|t|)
## (Intercept)  0.01775      0.00000    0.01801    0.99   0.32484
## L30_1        0.16678      0.89918    0.00360   46.29   < 2e-16
## L30_2        0.02266      0.08580    0.00493    4.59 0.0000056
## L30_3        0.02242      0.06815    0.00632    3.55   0.00043
##
## Residual standard error: 0.402 on 496 degrees of freedom
## Multiple R-squared:  0.831,Adjusted R-squared:  0.83
## F-statistic:  811 on 3 and 496 DF,  p-value: <2e-16
```

```
LM_4 <- lm(PC$V10 ~ L50_1 + L50_2 + L50_3, data = PC)
summary(LM_4)
```

```
##
## Call:
## lm(formula = PC$V10 ~ L50_1 + L50_2 + L50_3, data = PC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4161 -0.3076 -0.0216  0.3507  1.2573
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01194    0.02167    0.55     0.58
## L50_1        0.22737    0.00623   36.51   <2e-16
## L50_2        0.01336    0.00830    1.61     0.11
## L50_3        0.16220    0.01143   14.19   <2e-16
##
## Residual standard error: 0.484 on 496 degrees of freedom
## Multiple R-squared:  0.755,Adjusted R-squared:  0.753
## F-statistic:  509 on 3 and 496 DF,  p-value: <2e-16
```

```
summary(lm.beta(LM_4))
```

```
##
## Call:
## lm(formula = PC$V10 ~ L50_1 + L50_2 + L50_3, data = PC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4161 -0.3076 -0.0216  0.3507  1.2573
##
## Coefficients:
##             Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept)  0.01194      0.00000    0.02167    0.55     0.58
## L50_1        0.22737      0.94863    0.00623   36.51   <2e-16
## L50_2        0.01336      0.04150    0.00830    1.61     0.11
## L50_3        0.16220      0.40400    0.01143   14.19   <2e-16
##
## Residual standard error: 0.484 on 496 degrees of freedom
## Multiple R-squared:  0.755,Adjusted R-squared:  0.753
## F-statistic:  509 on 3 and 496 DF,  p-value: <2e-16
```

Predicting a 10th variable, the magnitude (and sometimes the sign) of prediction is not preserved for smaller components.

3 Group Contamination

The use of principal components analysis has a hidden danger when used in experimental research. Numerous measures might be collected and principal components analysis might seem to be a reasonable way to simplify the data prior to conducting major analyses.

In experimental data, however, treatment-induced mean differences can impose a structure on the data that may distort a principal components analysis attempting to uncover the underlying dimensionality of the outcome measures. This artificial systematic variability should be removed prior to conducting the PCA. This requires extracting the residuals for analysis. First let's see what happens if we ignore the group effects.

3.1 Data File

To examine the consequences of group contamination on principal components analysis, we'll use a data set (N = 500) in which participants are randomly assigned to one of two groups and 20 different outcome measures are collected.

```
setwd("C:\\Courses\\Psychology 516\\PowerPoint\\2018")
PC_2 <- read.table("groups_contamination_in_principal_components.csv",
    sep = ",", header = TRUE)
PC_2 <- as.data.frame(PC_2)
```

3.2 No Adjustment to the Data

3.2.1 Descriptive Statistics

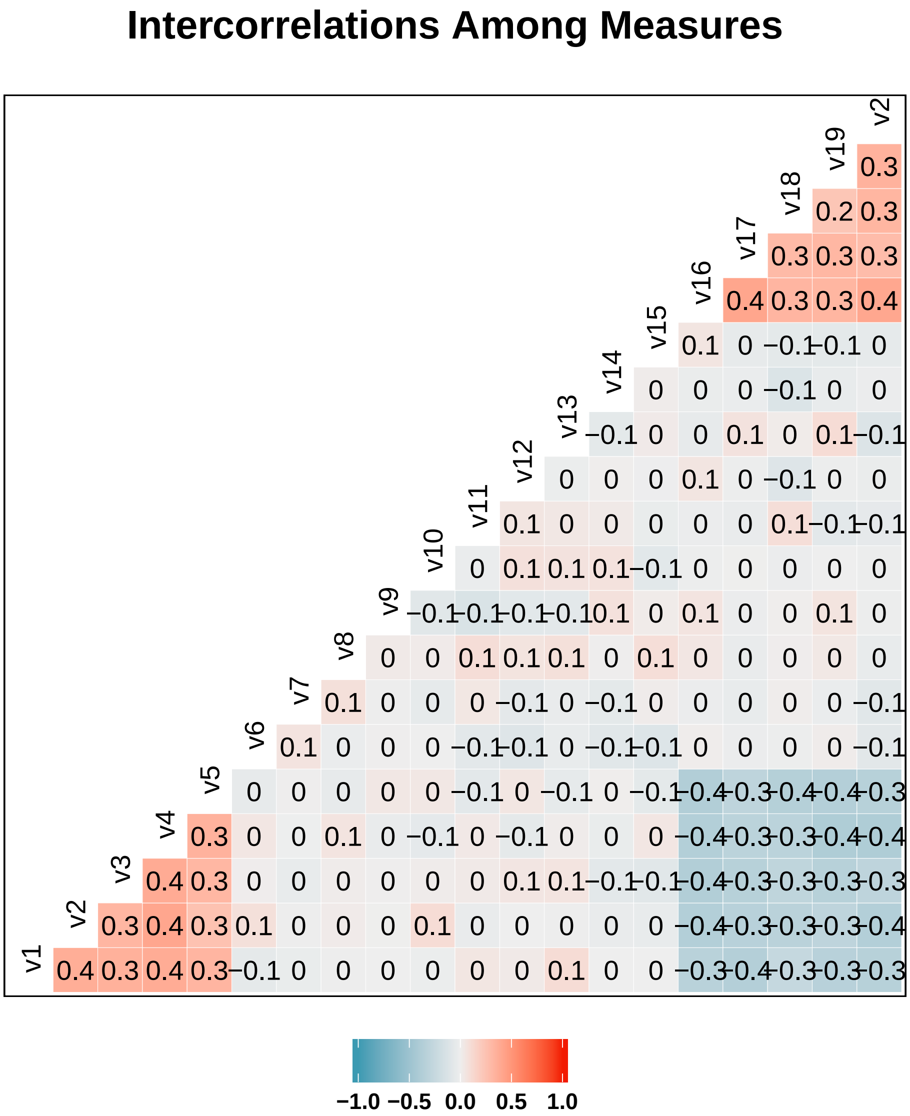
```
describe(PC_2[, c(1:20)])

##     vars   n  mean   sd median trimmed  mad   min  max range
## v1     1 500 -0.77 1.25  -0.75   -0.78 1.33 -4.22 2.73  6.94
## v2     2 500 -0.74 1.30  -0.73   -0.76 1.42 -4.21 3.37  7.58
## v3     3 500 -0.68 1.26  -0.61   -0.66 1.31 -5.11 2.43  7.54
## v4     4 500 -0.87 1.27  -0.87   -0.88 1.26 -4.22 3.78  8.00
## v5     5 500 -0.64 1.29  -0.64   -0.65 1.32 -4.11 2.88  6.99
## v6     6 500  0.01 1.00  -0.02    0.01 1.00 -3.03 2.75  5.78
## v7     7 500  0.07 1.01   0.06    0.07 0.97 -2.96 3.13  6.10
## v8     8 500 -0.02 1.05  -0.07   -0.02 0.99 -2.79 2.85  5.63
## v9     9 500  0.09 0.97   0.07    0.05 0.84 -2.18 2.67  4.86
## v10   10 500 -0.08 0.93  -0.11   -0.09 0.92 -2.62 2.35  4.97
## v11   11 500 -0.02 0.98  -0.01   -0.01 1.01 -3.71 2.33  6.04
## v12   12 500 -0.06 1.01  -0.11   -0.06 1.00 -3.12 2.57  5.69
## v13   13 500 -0.08 1.01  -0.17   -0.10 0.96 -2.64 3.20  5.83
## v14   14 500 -0.02 1.03  -0.02   -0.02 1.08 -2.54 2.71  5.26
## v15   15 500 -0.03 1.07  -0.03   -0.03 1.15 -2.53 2.78  5.31
## v16   16 500  0.73 1.19   0.70    0.72 1.26 -2.66 4.50  7.17
## v17   17 500  0.78 1.31   0.76    0.78 1.37 -2.92 4.90  7.81
## v18   18 500  0.69 1.29   0.72    0.69 1.34 -3.07 4.05  7.11
## v19   19 500  0.82 1.34   0.83    0.82 1.38 -2.98 4.56  7.54
## v20   20 500  0.72 1.23   0.70    0.73 1.28 -3.39 3.66  7.05
##      skew kurtosis   se
## v1   0.04    -0.34 0.06
## v2   0.13    -0.18 0.06
## v3  -0.13    -0.19 0.06
## v4   0.11    -0.19 0.06
## v5  -0.01    -0.33 0.06
## v6  -0.04    -0.01 0.04
## v7  -0.07     0.13 0.05
## v8   0.00     0.11 0.05
## v9   0.34    -0.14 0.04
## v10  0.06    -0.28 0.04
## v11 -0.23     0.21 0.04
## v12 -0.02    -0.04 0.05
## v13  0.22     0.14 0.05
## v14  0.04    -0.40 0.05
## v15  0.03    -0.53 0.05
## v16  0.10    -0.19 0.05
## v17  0.02    -0.31 0.06
## v18 -0.03    -0.38 0.06
## v19 -0.01    -0.22 0.06
## v20 -0.13    -0.31 0.06
```

```
R <- cor(PC_2[, c(1:20)])
round(R, 2)
```

```
##        v1    v2    v3    v4    v5    v6    v7    v8    v9   v10
## v1   1.00  0.37  0.35  0.38  0.33 -0.06 -0.03 -0.01  0.00 -0.02
## v2   0.37  1.00  0.32  0.41  0.26  0.08 -0.01  0.02  0.00  0.11
## v3   0.35  0.32  1.00  0.38  0.32  0.01 -0.04  0.02  0.00  0.01
## v4   0.38  0.41  0.38  1.00  0.34  0.04  0.00  0.06 -0.03 -0.06
## v5   0.33  0.26  0.32  0.34  1.00 -0.05  0.01 -0.04  0.04  0.04
## v6  -0.06  0.08  0.01  0.04 -0.05  1.00  0.06 -0.02  0.00  0.00
## v7  -0.03 -0.01 -0.04  0.00  0.01  0.06  1.00  0.08 -0.01 -0.05
## v8  -0.01  0.02  0.02  0.06 -0.04 -0.02  0.08  1.00  0.03  0.02
## v9   0.00  0.00  0.00 -0.03  0.04  0.00 -0.01  0.03  1.00 -0.08
## v10 -0.02  0.11  0.01 -0.06  0.04  0.00 -0.05  0.02 -0.08  1.00
## v11  0.05 -0.03  0.03  0.03 -0.07 -0.07  0.04  0.10 -0.13 -0.02
## v12  0.03  0.00  0.05 -0.06  0.05 -0.10 -0.06  0.06 -0.07  0.08
## v13  0.10 -0.01  0.06  0.01 -0.05 -0.04 -0.03  0.08 -0.07  0.07
## v14  0.00 -0.03 -0.07 -0.03  0.01 -0.08 -0.06  0.00  0.08  0.07
## v15  0.00 -0.04 -0.08  0.04 -0.06 -0.10  0.01  0.09  0.02 -0.07
## v16 -0.33 -0.37 -0.37 -0.37 -0.37  0.01 -0.02  0.05  0.06 -0.01
## v17 -0.36 -0.34 -0.34 -0.32 -0.30 -0.02 -0.04 -0.03 -0.02  0.00
## v18 -0.25 -0.32 -0.29 -0.31 -0.36 -0.02  0.01  0.01  0.01 -0.02
## v19 -0.34 -0.30 -0.34 -0.39 -0.36  0.02 -0.02  0.04  0.06  0.00
## v20 -0.34 -0.37 -0.30 -0.39 -0.34 -0.08 -0.08 -0.04 -0.01 -0.01
##       v11   v12   v13   v14   v15   v16   v17   v18   v19   v20
## v1   0.05  0.03  0.10  0.00  0.00 -0.33 -0.36 -0.25 -0.34 -0.34
## v2  -0.03  0.00 -0.01 -0.03 -0.04 -0.37 -0.34 -0.32 -0.30 -0.37
## v3   0.03  0.05  0.06 -0.07 -0.08 -0.37 -0.34 -0.29 -0.34 -0.30
## v4   0.03 -0.06  0.01 -0.03  0.04 -0.37 -0.32 -0.31 -0.39 -0.39
## v5  -0.07  0.05 -0.05  0.01 -0.06 -0.37 -0.30 -0.36 -0.36 -0.34
## v6  -0.07 -0.10 -0.04 -0.08 -0.10  0.01 -0.02 -0.02  0.02 -0.08
## v7   0.04 -0.06 -0.03 -0.06  0.01 -0.02 -0.04  0.01 -0.02 -0.08
## v8   0.10  0.06  0.08  0.00  0.09  0.05 -0.03  0.01  0.04 -0.04
## v9  -0.13 -0.07 -0.07  0.08  0.02  0.06 -0.02  0.01  0.06 -0.01
## v10 -0.02  0.08  0.07  0.07 -0.07 -0.01  0.00 -0.02  0.00 -0.01
## v11  1.00  0.05  0.04  0.03 -0.03 -0.02 -0.03  0.09 -0.07 -0.05
## v12  0.05  1.00 -0.02  0.01 -0.01  0.05 -0.01 -0.10 -0.01 -0.03
## v13  0.04 -0.02  1.00 -0.06  0.03 -0.04  0.07  0.02  0.10 -0.11
## v14  0.03  0.01 -0.06  1.00  0.01 -0.03 -0.02 -0.12 -0.04 -0.02
## v15 -0.03 -0.01  0.03  0.01  1.00  0.05 -0.05 -0.06 -0.06 -0.05
## v16 -0.02  0.05 -0.04 -0.03  0.05  1.00  0.41  0.33  0.32  0.41
## v17 -0.03 -0.01  0.07 -0.02 -0.05  0.41  1.00  0.30  0.32  0.29
## v18  0.09 -0.10  0.02 -0.12 -0.06  0.33  0.30  1.00  0.23  0.32
## v19 -0.07 -0.01  0.10 -0.04 -0.06  0.32  0.32  0.23  1.00  0.34
## v20 -0.05 -0.03 -0.11 -0.02 -0.05  0.41  0.29  0.32  0.34  1.00
```

A heat map for the correlation matrix suggests clear patterning in the data.

```r
ggcorr(PC_2[, c(1:20)], label = TRUE, angle = 90, hjust = 0.1, size = 4,
    digits = 2) + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Intercorrelations Among Measures")
```



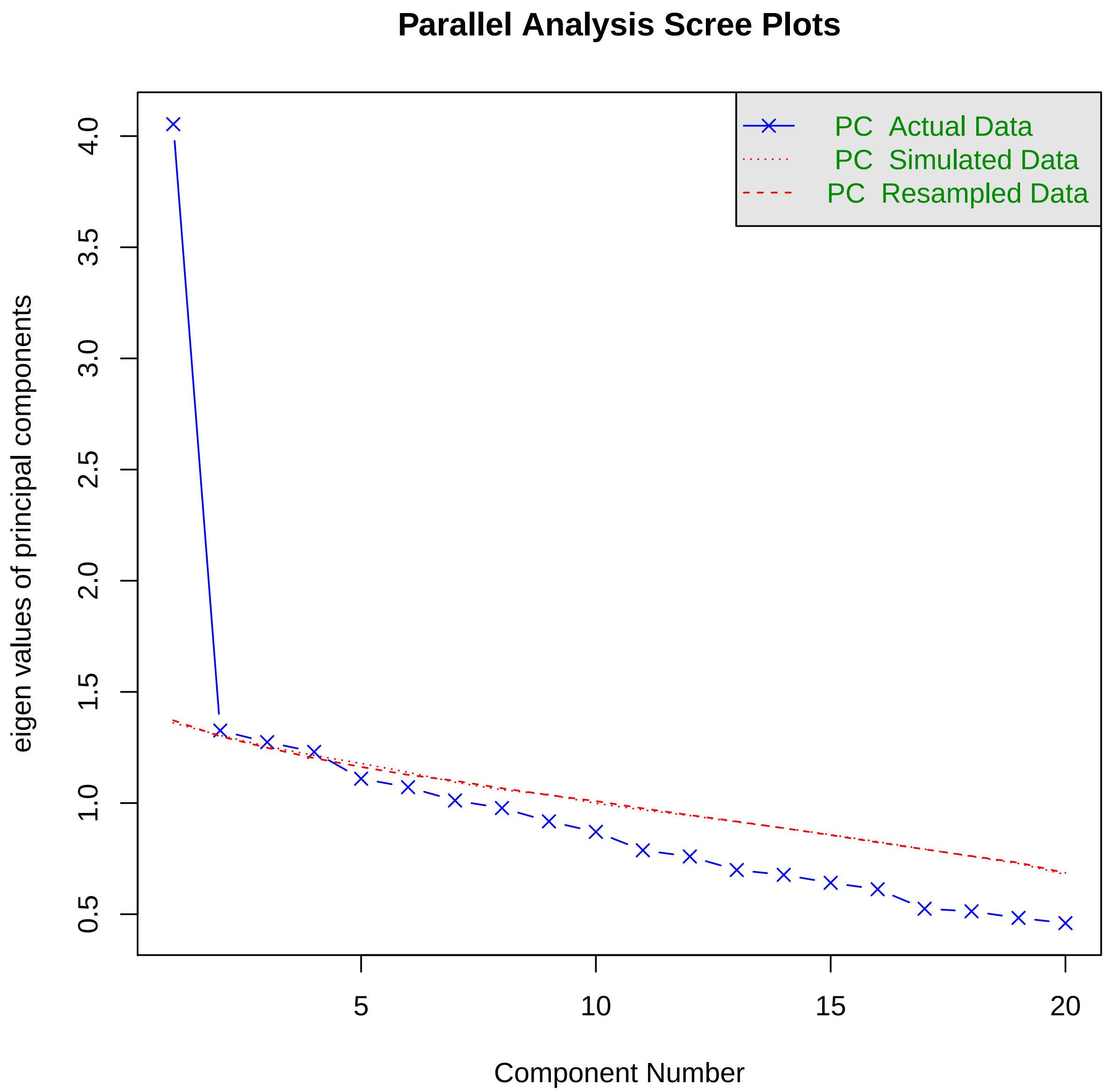
3.2.2 Should a PCA be Conducted?

```
KMO(R)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = R)
## Overall MSA =  0.84
## MSA for each item =
##   v1   v2   v3   v4   v5   v6   v7   v8   v9  v10  v11  v12  v13
## 0.89 0.88 0.89 0.89 0.87 0.45 0.47 0.46 0.41 0.42 0.47 0.44 0.40
##  v14  v15  v16  v17  v18  v19  v20
## 0.40 0.38 0.88 0.89 0.86 0.87 0.88
```

```
cortest.bartlett(R = R, n = 500)

## $chisq
## [1] 1447
##
## $p.value
## [1] 4.029e-192
##
## $df
## [1] 190
```

3.2.3 Scree Test

```
scree <- fa.parallel(PC_2[, c(1:20)], fa = "pc")
```



```
## Parallel analysis suggests that the number of factors =  NA  and the number of components =  4
```

The KMO, Bartlett, and scree tests all suggest that a principal components analysis should be done. There appears to be a single dominant component in the data. If true, this would imply that the 20 measures might be replaced by one in subsequent analyses.

3.2.4 PCA

```
PCA_2 <- principal(R, nfactors = 1, rotate = "none", n.obs = 500,
    residuals = TRUE)
PCA_2

## Principal Components Analysis
## Call: principal(r = R, nfactors = 1, residuals = TRUE, rotate = "none",
##     n.obs = 500)
## Standardized loadings (pattern matrix) based upon correlation matrix
##       PC1       h2   u2 com
## v1   0.64 0.405669 0.59   1
## v2   0.64 0.408551 0.59   1
## v3   0.63 0.399740 0.60   1
## v4   0.68 0.463356 0.54   1
## v5   0.62 0.385216 0.61   1
## v6   0.02 0.000483 1.00   1
## v7   0.01 0.000218 1.00   1
## v8   0.01 0.000061 1.00   1
## v9  -0.02 0.000403 1.00   1
## v10  0.03 0.000793 1.00   1
## v11  0.02 0.000523 1.00   1
## v12  0.03 0.000949 1.00   1
## v13  0.02 0.000433 1.00   1
## v14  0.02 0.000299 1.00   1
## v15  0.00 0.000014 1.00   1
## v16 -0.68 0.461628 0.54   1
## v17 -0.62 0.387380 0.61   1
## v18 -0.58 0.331961 0.67   1
## v19 -0.62 0.382573 0.62   1
## v20 -0.65 0.423010 0.58   1
##
##                  PC1
## SS loadings    4.05
## Proportion Var 0.20
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.06
##  with the empirical chi square  610.5  with prob <  5.8e-51
##
## Fit based upon off diagonal values = 0.89
```

3.2.5 Examination of Residuals

A residual matrix gives the variances in the main diagonal and correlations in the off-diagonals. This can be converted to a correlation matrix, which can then be examined using the KMO and Bartlett tests to determine if additional components should be extracted.

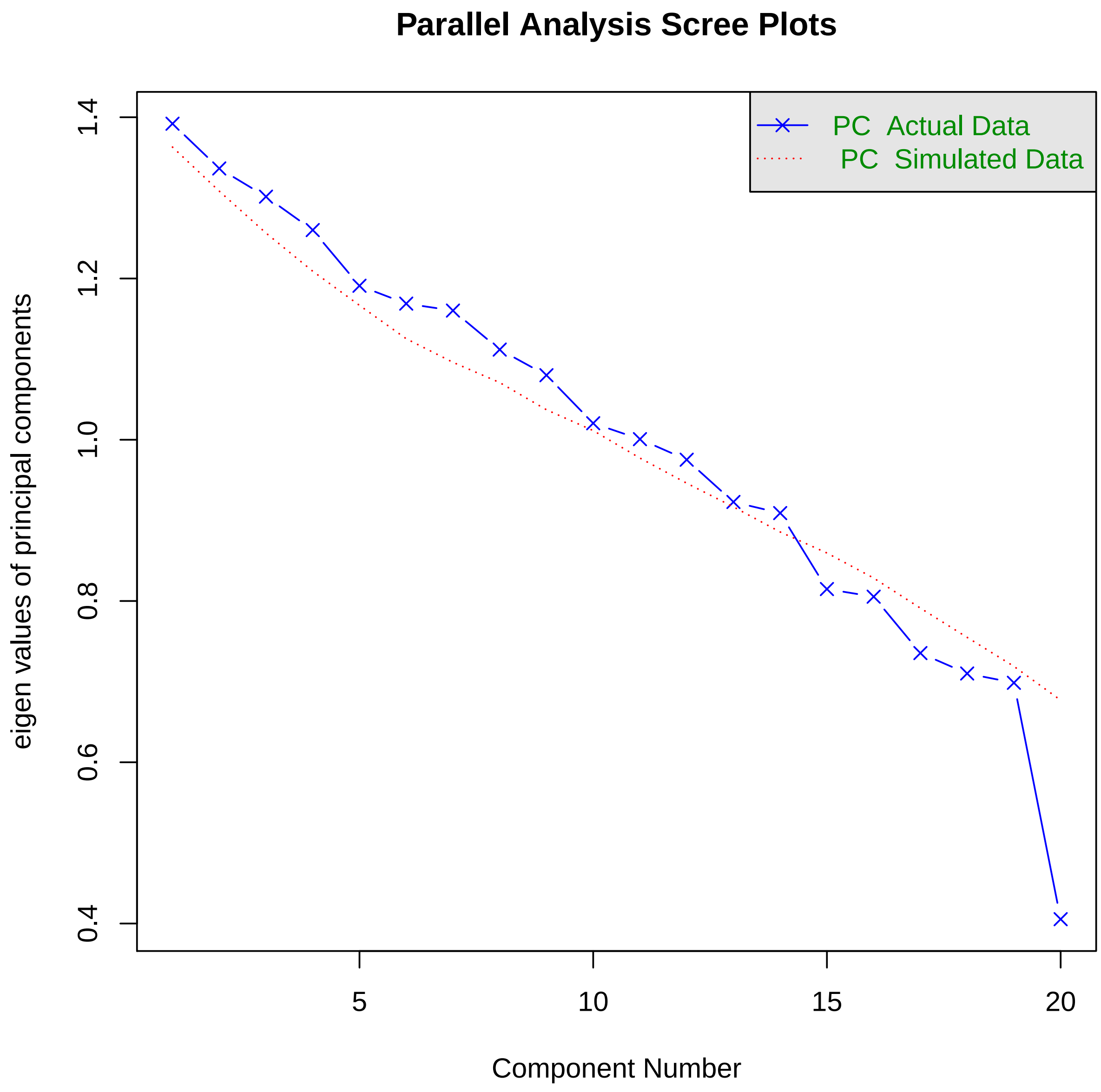
```
# Create a correlation matrix of the residuals by replacing the
# main diagonal with ones.
R1 <- diag(PCA_2$residual)
R2 <- diag(R1)
R3 <- PCA_2$residual - R2
R4 <- diag(20) + R3

# Assess the factorability of the residual correlation matrix.
KMO(R4)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = R4)
## Overall MSA =  0.34
## MSA for each item =
##   v1   v2   v3   v4   v5   v6   v7   v8   v9  v10  v11  v12  v13
## 0.25 0.27 0.25 0.26 0.28 0.52 0.53 0.51 0.48 0.50 0.52 0.52 0.52
##  v14  v15  v16  v17  v18  v19  v20
## 0.49 0.49 0.25 0.25 0.30 0.28 0.28
```

```
cortest.bartlett(R = R4, n = length(Data[, 1]))
```

```
## $chisq
## [1] 361
##
## $p.value
## [1] 9.986e-13
##
## $df
## [1] 190
```

```
scree <- fa.parallel(R4, fa = "pc", n.obs = length(Data[, 1]))
```



```
## Parallel analysis suggests that the number of factors =  NA  and the number of components =  14
```

No evidence of additional components.

3.3 Adjust the Data for Group Differences

Because the data came from an experiment, there is probably variation in the scores that is due to the manipulation. That variation could be artificially inflating or deflating the correlations among the variables. It needs to be removed before a principal components analysis is conducted.

3.3.1 Group Differences: t-tests

```
# t-tests to determine if group differences exist.
t.test(v1 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v1 by group
## t = -17, df = 500, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.671 -1.316
## sample estimates:
## mean in group 1 mean in group 2
##        -1.51543        -0.02174
```

```
t.test(v2 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v2 by group
## t = -16, df = 500, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.689 -1.314
## sample estimates:
## mean in group 1 mean in group 2
##       -1.492273        0.009125
```

```
t.test(v3 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v3 by group
## t = -16, df = 500, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.675 -1.319
## sample estimates:
## mean in group 1 mean in group 2
##        -1.42231         0.07464
```

```
t.test(v4 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v4 by group
## t = -16, df = 500, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.669 -1.306
## sample estimates:
## mean in group 1 mean in group 2
##         -1.6130         -0.1259
```

```r
t.test(v5 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v5 by group
## t = -16, df = 500, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.680 -1.309
## sample estimates:
## mean in group 1 mean in group 2
##         -1.3889          0.1057
```

```r
t.test(v6 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v6 by group
## t = 0.16, df = 500, p-value = 0.9
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1610  0.1901
## sample estimates:
## mean in group 1 mean in group 2
##        0.018491        0.003924
```

```r
t.test(v7 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v7 by group
## t = 0.11, df = 500, p-value = 0.9
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1672  0.1878
## sample estimates:
## mean in group 1 mean in group 2
##         0.07053         0.06021
```

```r
t.test(v8 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v8 by group
## t = -0.045, df = 500, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1883  0.1798
## sample estimates:
## mean in group 1 mean in group 2
##        -0.02371        -0.01946
```

```r
t.test(v9 ~ group, data = PC_2)

##
##  Welch Two Sample t-test
##
## data:  v9 by group
## t = 0.074, df = 500, p-value = 0.9
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1649  0.1778
## sample estimates:
## mean in group 1 mean in group 2
##         0.09059         0.08411

t.test(v10 ~ group, data = PC_2)

##
##  Welch Two Sample t-test
##
## data:  v10 by group
## t = -0.035, df = 500, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1673  0.1615
## sample estimates:
## mean in group 1 mean in group 2
##        -0.08155        -0.07866

t.test(v11 ~ group, data = PC_2)

##
##  Welch Two Sample t-test
##
## data:  v11 by group
## t = 0.023, df = 500, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1702  0.1742
## sample estimates:
## mean in group 1 mean in group 2
##        -0.02099        -0.02304

t.test(v12 ~ group, data = PC_2)

##
##  Welch Two Sample t-test
##
## data:  v12 by group
## t = -0.19, df = 500, p-value = 0.9
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1953  0.1612
## sample estimates:
## mean in group 1 mean in group 2
##        -0.07280        -0.05576
```

```
t.test(v13 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v13 by group
## t = -0.028, df = 500, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1796  0.1745
## sample estimates:
## mean in group 1 mean in group 2
##        -0.08232        -0.07980
```

```
t.test(v14 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v14 by group
## t = 0.026, df = 500, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1795  0.1844
## sample estimates:
## mean in group 1 mean in group 2
##        -0.01649        -0.01893
```

```
t.test(v15 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v15 by group
## t = 0.12, df = 500, p-value = 0.9
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1762  0.1988
## sample estimates:
## mean in group 1 mean in group 2
##        -0.02227        -0.03361
```

```
t.test(v16 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v16 by group
## t = 18, df = 500, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.330 1.655
## sample estimates:
## mean in group 1 mean in group 2
##          1.4726         -0.0197
```

```
t.test(v17 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v17 by group
## t = 16, df = 500, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.311 1.691
## sample estimates:
## mean in group 1 mean in group 2
##          1.5248          0.0241
```

```
t.test(v18 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v18 by group
## t = 16, df = 500, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.308 1.679
## sample estimates:
## mean in group 1 mean in group 2
##         1.43076        -0.06294
```

```
t.test(v19 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v19 by group
## t = 15, df = 500, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.291 1.682
## sample estimates:
## mean in group 1 mean in group 2
##          1.5564          0.0698
```

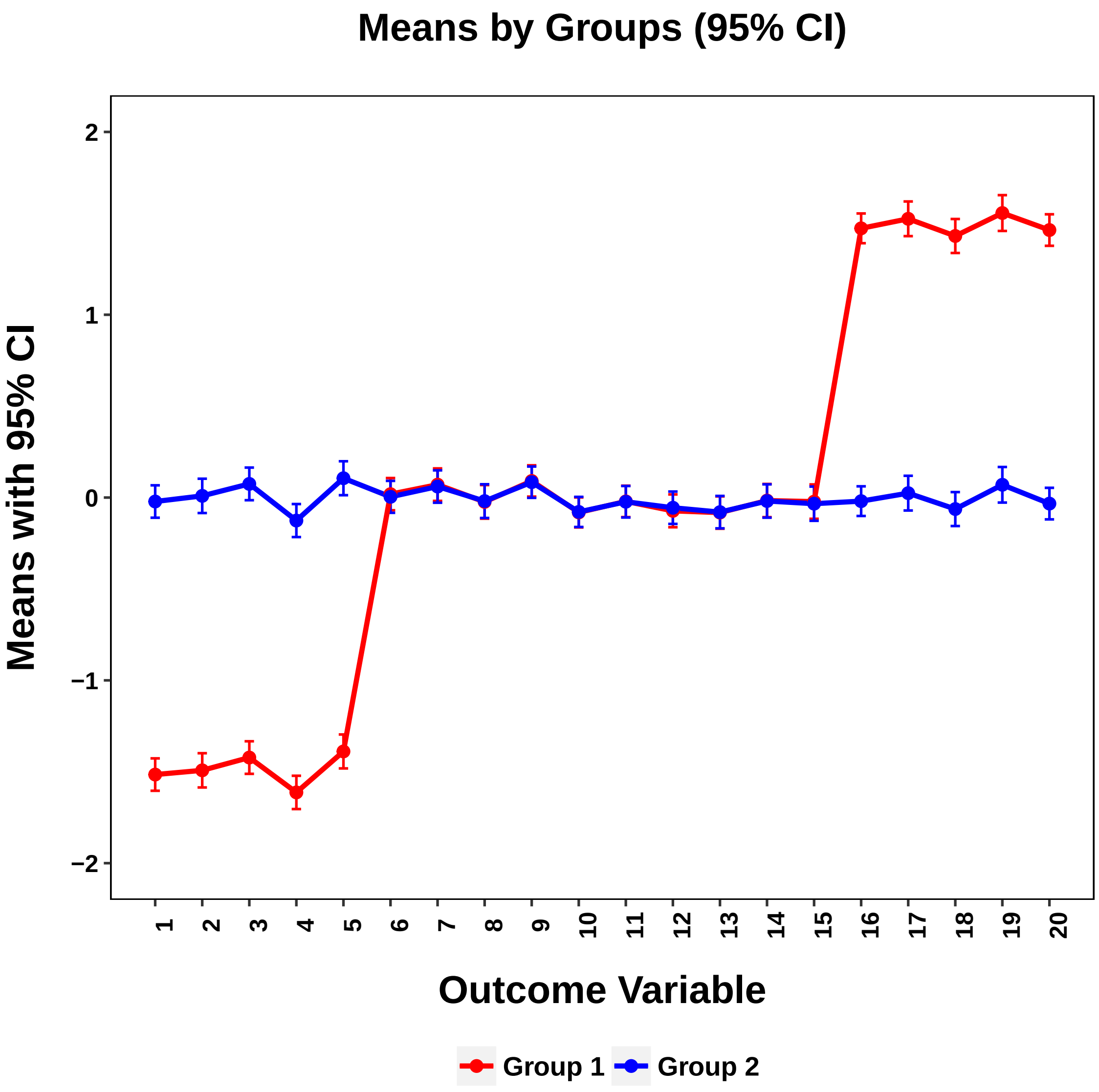
```
t.test(v20 ~ group, data = PC_2)
```

```
##
##  Welch Two Sample t-test
##
## data:  v20 by group
## t = 17, df = 500, p-value <2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.324 1.669
## sample estimates:
## mean in group 1 mean in group 2
##          1.4633         -0.0331
```

3.3.2 Graphical Display of Group Differences

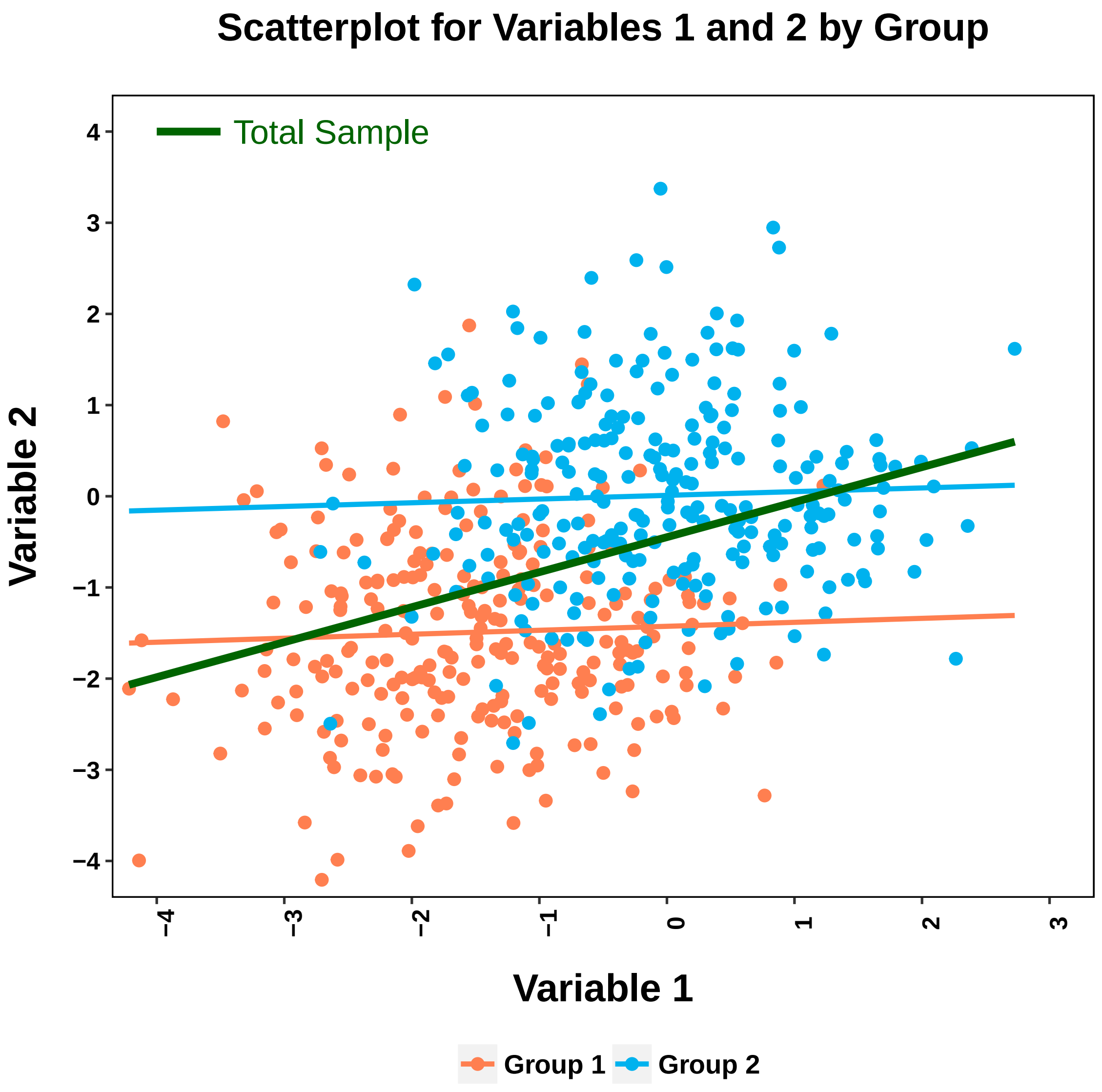
```
# Graphical display of group differences.
means <- aggregate(v1 ~ group, PC_2, mean)
means <- cbind(means, aggregate(v2 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v3 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v4 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v5 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v6 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v7 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v8 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v9 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v10 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v11 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v12 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v13 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v14 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v15 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v16 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v17 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v18 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v19 ~ group, PC_2, mean)[, 2])
means <- cbind(means, aggregate(v20 ~ group, PC_2, mean)[, 2])
means <- as.data.frame(means)
names(means) <- c("groups", "v1", "v2", "v3", "v4", "v5", "v6", "v7",
    "v8", "v9", "v10", "v11", "v12", "v13", "v14", "v15", "v16", "v17",
    "v18", "v19", "v20")

sds <- aggregate(v1 ~ group, PC_2, sd)
sds <- cbind(sds, aggregate(v2 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v3 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v4 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v5 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v6 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v7 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v8 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v9 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v10 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v11 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v12 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v13 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v14 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v15 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v16 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v17 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v18 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v19 ~ group, PC_2, sd)[, 2])
sds <- cbind(sds, aggregate(v20 ~ group, PC_2, sd)[, 2])
sds <- as.data.frame(sds)
names(sds) <- c("groups", "v1", "v2", "v3", "v4", "v5", "v6", "v7",
    "v8", "v9", "v10", "v11", "v12", "v13", "v14", "v15", "v16", "v17",
    "v18", "v19", "v20")

means_long <- melt(means, id.vars = "groups")
means_long <- as.data.frame(means_long)
names(means_long) <- c("groups", "variable", "means")
sds_long <- melt(sds, id.vars = "groups")
sds_long <- as.data.frame(sds_long)
names(sds_long) <- c("groups", "variable", "sds")
plot_data <- cbind(means_long, sds_long[, 3])
plot_data <- as.data.frame(plot_data)
names(plot_data) <- c("groups", "variable", "means", "sds")
plot_data$CI_Low <- plot_data$means - qt(0.975, df = length(PC_2[,
    1]) - 1, lower.tail = TRUE) * plot_data$sds/sqrt(length(PC_2[,
    1]))
plot_data$CI_High <- plot_data$means + qt(0.975, df = length(PC_2[,
    1]) - 1, lower.tail = TRUE) * plot_data$sds/sqrt(length(PC_2[,
    1]))
plot_data$groups_F <- factor(plot_data$groups, levels = c(1, 2), labels = c("Group 1",
    "Group 2"))
plot_data$variable_2 <- c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7,
    8, 8, 9, 9, 10, 10, 11, 11, 12, 12, 13, 13, 14, 14, 15, 15, 16,
    16, 17, 17, 18, 18, 19, 19, 20, 20)
```

```
ggplot(plot_data, aes(x = variable_2, y = means, color = groups_F)) +
    geom_errorbar(aes(ymin = CI_Low, ymax = CI_High), width = 0.2) +
    geom_line(size = 1) + geom_point(size = 2) + scale_color_manual(values = c("red",
    "blue")) + scale_y_continuous(breaks = c(-2, -1, 0, 1, 2)) + scale_x_continuous(breaks = seq(1,
    20, 1)) + coord_cartesian(xlim = c(1, 20), ylim = c(-2, 2)) +
    xlab("Outcome Variable") + ylab("Means with 95% CI") + theme(text = element_text(size = 14,
    family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 10, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 10, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Means by Groups (95% CI)")
```

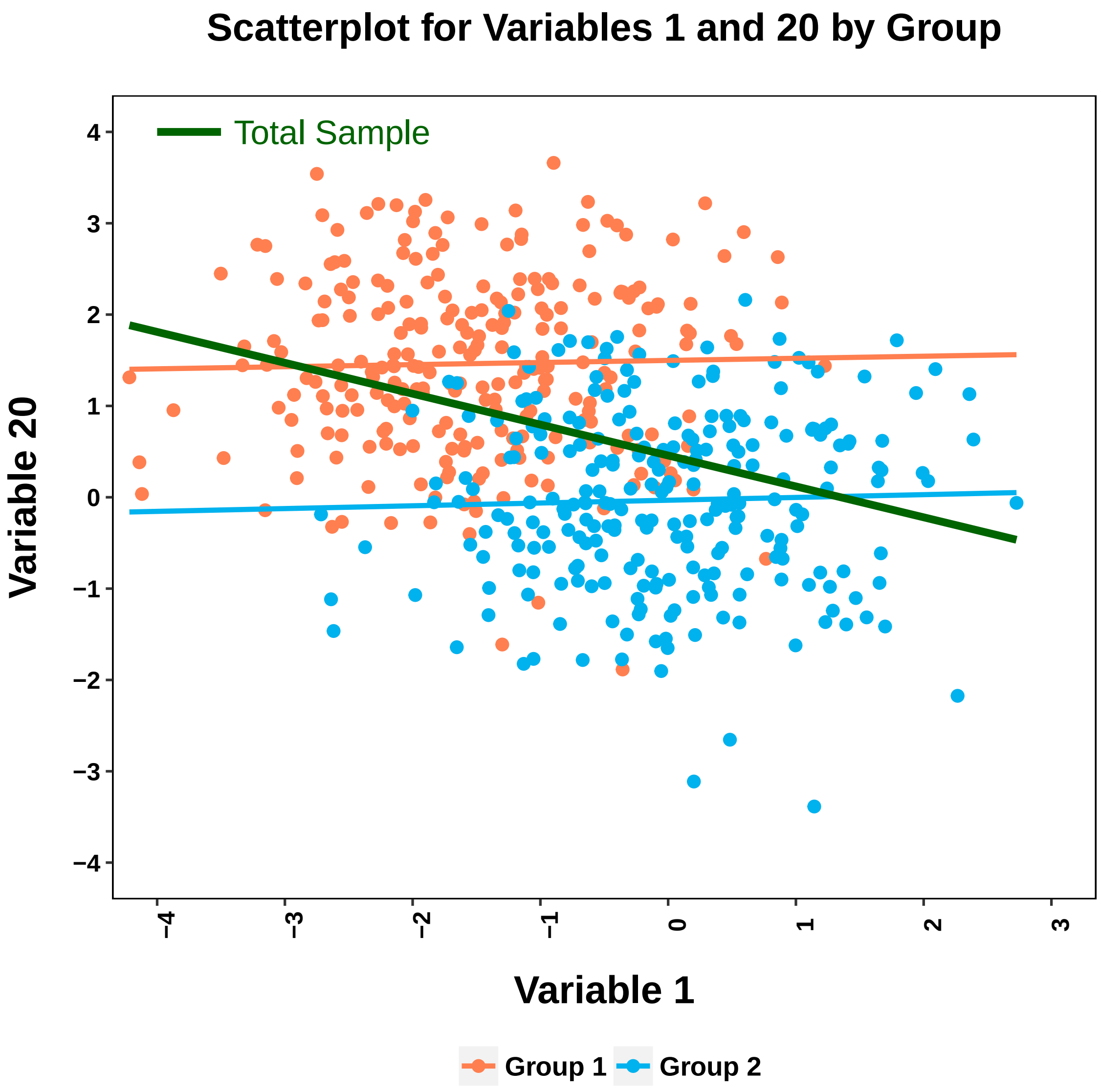


3.3.3 Illustration of How Group Differences Influence Correlations

```
PC_2$groups_F <- factor(PC_2$group, levels = c(1, 2), labels = c("Group 1",
    "Group 2"))
ggplot(PC_2, aes(x = v1, y = v2, color = groups_F)) + geom_point(size = 2) +
    geom_smooth(method = lm, se = FALSE, fullrange = TRUE, size = 1) +
    geom_smooth(aes(x = v1, y = v2), method = lm, se = FALSE, color = "darkgreen",
        size = 1.5, linetype = 1) + scale_color_manual(values = c("coral",
    "deepskyblue2")) + scale_y_continuous(breaks = seq(-4, 4, 1)) +
    scale_x_continuous(breaks = seq(-4, 3, 1)) + coord_cartesian(xlim = c(-4,
    3), ylim = c(-4, 4)) + xlab("Variable 1") + ylab("Variable 2") +
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
    size = 10, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 10, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + annotate("text", label = "Total Sample",
x = -3.4, y = 4, color = "darkgreen", hjust = 0, size = 5) + annotate("segment",
x = -4, xend = -3.5, y = 4, yend = 4, color = "darkgreen", size = 1.5) +
ggtitle("Scatterplot for Variables 1 and 2 by Group")
```



```
ggplot(PC_2, aes(x = v1, y = v20, color = groups_F)) + geom_point(size = 2) +
    geom_smooth(method = lm, se = FALSE, fullrange = TRUE, size = 1) +
    geom_smooth(aes(x = v1, y = v20), method = lm, se = FALSE, color = "darkgreen",
        size = 1.5, linetype = 1) + scale_color_manual(values = c("coral",
    "deepskyblue2")) + scale_y_continuous(breaks = seq(-4, 4, 1)) +
    scale_x_continuous(breaks = seq(-4, 3, 1)) + coord_cartesian(xlim = c(-4,
    3), ylim = c(-4, 4)) + xlab("Variable 1") + ylab("Variable 20") +
    theme(text = element_text(size = 14, family = "sans", color = "black",
        face = "bold"), axis.text.y = element_text(colour = "black",
        size = 10, face = "bold"), axis.text.x = element_text(colour = "black",
        size = 10, face = "bold", angle = 90), axis.title.x = element_text(margin = margin(15,
        0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
        15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
        plot.title = element_text(size = 16, face = "bold", margin = margin(0,
            0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
            linetype = 1, color = "black"), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
        plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
        legend.title = element_blank()) + annotate("text", label = "Total Sample",
    x = -3.4, y = 4, color = "darkgreen", hjust = 0, size = 5) + annotate("segment",
    x = -4, xend = -3.5, y = 4, yend = 4, color = "darkgreen", size = 1.5) +
    ggtitle("Scatterplot for Variables 1 and 20 by Group")
```



3.3.4 Linear Models to Remove Group Differences

A separate linear model, using group as a predictor, is estimated. The residuals from those analyses are then collected in one file for further analyses.

```
M_1 <- lm(v1 ~ group, data = PC_2)
M_2 <- lm(v2 ~ group, data = PC_2)
M_3 <- lm(v3 ~ group, data = PC_2)
M_4 <- lm(v4 ~ group, data = PC_2)
M_5 <- lm(v5 ~ group, data = PC_2)
M_6 <- lm(v6 ~ group, data = PC_2)
M_7 <- lm(v7 ~ group, data = PC_2)
M_8 <- lm(v8 ~ group, data = PC_2)
M_9 <- lm(v9 ~ group, data = PC_2)
```

```
M_10 <- lm(v10 ~ group, data = PC_2)
M_11 <- lm(v11 ~ group, data = PC_2)
M_12 <- lm(v12 ~ group, data = PC_2)
M_13 <- lm(v13 ~ group, data = PC_2)
M_14 <- lm(v14 ~ group, data = PC_2)
M_15 <- lm(v15 ~ group, data = PC_2)
M_16 <- lm(v16 ~ group, data = PC_2)
M_17 <- lm(v17 ~ group, data = PC_2)
M_18 <- lm(v18 ~ group, data = PC_2)
M_19 <- lm(v19 ~ group, data = PC_2)
M_20 <- lm(v20 ~ group, data = PC_2)
```

3.3.5 Retain Residuals for Further Analyses

```
PC_R <- cbind(M_1$residuals, M_2$residuals, M_3$residuals, M_4$residuals,
    M_5$residuals, M_6$residuals, M_7$residuals, M_8$residuals, M_9$residuals,
    M_10$residuals, M_11$residuals, M_12$residuals, M_13$residuals,
    M_14$residuals, M_15$residuals, M_16$residuals, M_17$residuals,
    M_18$residuals, M_19$residuals, M_20$residuals)
PC_R <- as.data.frame(PC_R)
names(PC_R) <- c("v1", "v2", "v3", "v4", "v5", "v6", "v7", "v8", "v9",
    "v10", "v11", "v12", "v13", "v14", "v15", "v16", "v17", "v18",
    "v19", "v20")
```

3.4 Analyze Residuals

3.4.1 Descriptive Statistics

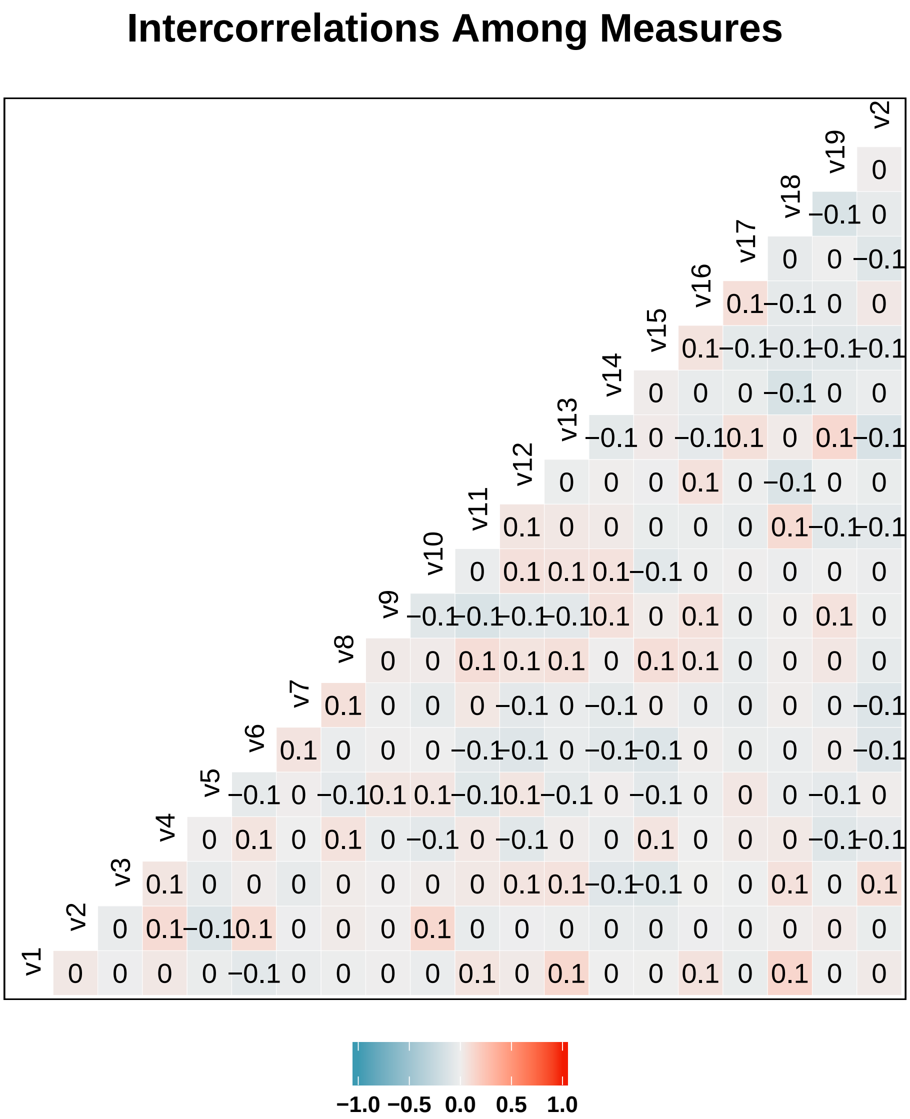
```
describe(PC_R)

##      vars   n mean   sd median trimmed  mad   min  max range
## v1     1 500    0 1.01  -0.03   -0.01 0.96 -2.70 2.75  5.45
## v2     2 500    0 1.07  -0.13   -0.02 0.98 -2.72 3.37  6.08
## v3     3 500    0 1.01   0.04    0.02 1.02 -3.69 2.36  6.04
## v4     4 500    0 1.03   0.01   -0.01 1.06 -2.60 3.90  6.51
## v5     5 500    0 1.06   0.01    0.00 1.03 -2.72 2.77  5.49
## v6     6 500    0 1.00  -0.03    0.00 1.01 -3.05 2.74  5.79
## v7     7 500    0 1.01   0.00    0.01 0.97 -3.03 3.07  6.11
## v8     8 500    0 1.05  -0.05    0.00 0.99 -2.77 2.87  5.64
## v9     9 500    0 0.97  -0.01   -0.04 0.84 -2.27 2.59  4.86
## v10   10 500    0 0.93  -0.03   -0.01 0.92 -2.54 2.43  4.97
## v11   11 500    0 0.98   0.01    0.02 1.01 -3.68 2.36  6.04
## v12   12 500    0 1.01  -0.05    0.00 1.00 -3.06 2.65  5.70
## v13   13 500    0 1.01  -0.09   -0.01 0.96 -2.56 3.28  5.84
## v14   14 500    0 1.03  -0.01    0.00 1.08 -2.53 2.73  5.26
## v15   15 500    0 1.07   0.00    0.00 1.14 -2.50 2.81  5.32
## v16   16 500    0 0.92   0.00   -0.02 0.92 -2.64 3.03  5.67
## v17   17 500    0 1.08   0.12    0.00 1.17 -2.94 3.37  6.32
## v18   18 500    0 1.06  -0.01    0.01 1.08 -3.00 2.61  5.62
## v19   19 500    0 1.11   0.05    0.01 1.12 -3.05 3.00  6.05
## v20   20 500    0 0.98  -0.02    0.01 1.09 -3.35 2.20  5.55
##      skew kurtosis   se
## v1   0.07    -0.19 0.05
## v2   0.24     0.11 0.05
## v3  -0.25     0.16 0.05
## v4   0.22     0.13 0.05
## v5  -0.02    -0.24 0.05
## v6  -0.04    -0.01 0.04
## v7  -0.07     0.13 0.05
## v8   0.00     0.11 0.05
## v9   0.34    -0.14 0.04
## v10  0.06    -0.28 0.04
## v11 -0.23     0.21 0.04
## v12 -0.02    -0.04 0.05
## v13  0.22     0.14 0.05
## v14  0.04    -0.40 0.05
## v15  0.03    -0.53 0.05
## v16  0.20     0.35 0.04
## v17  0.03    -0.19 0.05
## v18 -0.06    -0.35 0.05
## v19 -0.03    -0.04 0.05
## v20 -0.25    -0.08 0.04
```

```
R <- cor(PC_R)
round(R, 2)
```

```
##        v1    v2    v3    v4    v5    v6    v7    v8    v9   v10
## v1   1.00  0.04 -0.01  0.04 -0.03 -0.07 -0.03 -0.01  0.00 -0.02
## v2   0.04  1.00 -0.03  0.11 -0.11  0.11 -0.01  0.03  0.01  0.13
## v3  -0.01 -0.03  1.00  0.05 -0.04  0.02 -0.05  0.02  0.01  0.02
## v4   0.04  0.11  0.05  1.00  0.01  0.06  0.00  0.07 -0.04 -0.07
## v5  -0.03 -0.11 -0.04  0.01  1.00 -0.05  0.01 -0.05  0.05  0.05
## v6  -0.07  0.11  0.02  0.06 -0.05  1.00  0.06 -0.02  0.00  0.00
## v7  -0.03 -0.01 -0.05  0.00  0.01  0.06  1.00  0.08 -0.01 -0.05
## v8  -0.01  0.03  0.02  0.07 -0.05 -0.02  0.08  1.00  0.03  0.02
## v9   0.00  0.01  0.01 -0.04  0.05  0.00 -0.01  0.03  1.00 -0.08
## v10 -0.02  0.13  0.02 -0.07  0.05  0.00 -0.05  0.02 -0.08  1.00
## v11  0.06 -0.04  0.04  0.04 -0.08 -0.07  0.04  0.10 -0.13 -0.02
## v12  0.03 -0.01  0.06 -0.08  0.05 -0.10 -0.06  0.06 -0.07  0.08
## v13  0.13 -0.01  0.07  0.02 -0.06 -0.04 -0.03  0.08 -0.07  0.07
## v14  0.00 -0.04 -0.09 -0.03  0.01 -0.08 -0.06  0.00  0.08  0.07
## v15  0.00 -0.04 -0.10  0.06 -0.07 -0.10  0.01  0.09  0.02 -0.07
## v16  0.07 -0.01  0.00  0.00 -0.01  0.01 -0.03  0.06  0.07 -0.01
## v17 -0.03 -0.01  0.00  0.03  0.05 -0.03 -0.05 -0.04 -0.03  0.00
## v18  0.14  0.01  0.08  0.04 -0.03 -0.02  0.01  0.01  0.01 -0.02
## v19  0.00  0.03 -0.02 -0.09 -0.06  0.02 -0.03  0.05  0.07  0.00
## v20  0.03 -0.03  0.09 -0.05  0.01 -0.10 -0.11 -0.05 -0.02 -0.02
##       v11   v12   v13   v14   v15   v16   v17   v18   v19   v20
## v1   0.06  0.03  0.13  0.00  0.00  0.07 -0.03  0.14  0.00  0.03
## v2  -0.04 -0.01 -0.01 -0.04 -0.04 -0.01 -0.01  0.01  0.03 -0.03
## v3   0.04  0.06  0.07 -0.09 -0.10  0.00  0.00  0.08 -0.02  0.09
## v4   0.04 -0.08  0.02 -0.03  0.06  0.00  0.03  0.04 -0.09 -0.05
## v5  -0.08  0.05 -0.06  0.01 -0.07 -0.01  0.05 -0.03 -0.06  0.01
## v6  -0.07 -0.10 -0.04 -0.08 -0.10  0.01 -0.03 -0.02  0.02 -0.10
## v7   0.04 -0.06 -0.03 -0.06  0.01 -0.03 -0.05  0.01 -0.03 -0.11
## v8   0.10  0.06  0.08  0.00  0.09  0.06 -0.04  0.01  0.05 -0.05
## v9  -0.13 -0.07 -0.07  0.08  0.02  0.07 -0.03  0.01  0.07 -0.02
## v10 -0.02  0.08  0.07  0.07 -0.07 -0.01  0.00 -0.02  0.00 -0.02
## v11  1.00  0.05  0.04  0.03 -0.03 -0.03 -0.04  0.11 -0.08 -0.07
## v12  0.05  1.00 -0.02  0.01 -0.01  0.07 -0.01 -0.12 -0.01 -0.03
## v13  0.04 -0.02  1.00 -0.06  0.03 -0.06  0.08  0.02  0.12 -0.14
## v14  0.03  0.01 -0.06  1.00  0.01 -0.03 -0.03 -0.14 -0.05 -0.02
## v15 -0.03 -0.01  0.03  0.01  1.00  0.07 -0.06 -0.08 -0.08 -0.07
## v16 -0.03  0.07 -0.06 -0.03  0.07  1.00  0.09 -0.06 -0.04  0.04
## v17 -0.04 -0.01  0.08 -0.03 -0.06  0.09  1.00 -0.05  0.00 -0.09
## v18  0.11 -0.12  0.02 -0.14 -0.08 -0.06 -0.05  1.00 -0.13 -0.04
## v19 -0.08 -0.01  0.12 -0.05 -0.08 -0.04  0.00 -0.13  1.00  0.01
## v20 -0.07 -0.03 -0.14 -0.02 -0.07  0.04 -0.09 -0.04  0.01  1.00
```

A heat map for the correlation matrix suggests no clear patterning in the data.

```
ggcorr(PC_R[, c(1:20)], label = TRUE, angle = 90, hjust = 0.1, size = 4,
    digits = 2) + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Intercorrelations Among Measures")
```


Intercorrelations Among Measures

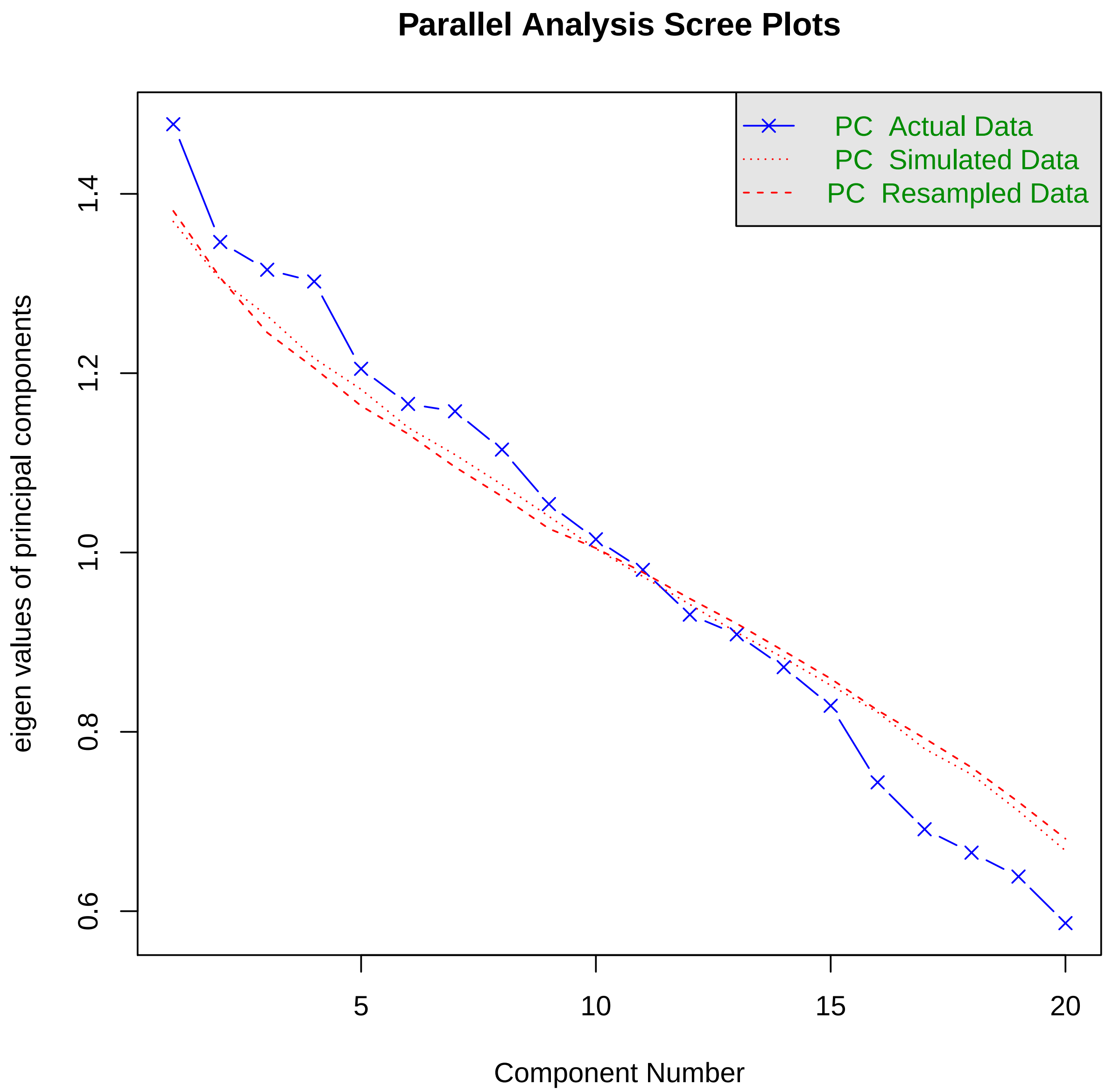
3.4.2 Should a PCA be Conducted?

```
KMO(R)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = R)
## Overall MSA =  0.46
## MSA for each item =
##   v1   v2   v3   v4   v5   v6   v7   v8   v9  v10  v11  v12  v13
## 0.46 0.46 0.46 0.48 0.46 0.48 0.50 0.49 0.44 0.46 0.53 0.45 0.48
##  v14  v15  v16  v17  v18  v19  v20
## 0.44 0.45 0.44 0.43 0.47 0.45 0.44
```

```
cortest.bartlett(R = R, n = 500)

## $chisq
## [1] 333.3
##
## $p.value
## [1] 6.208e-10
##
## $df
## [1] 190
```

3.4.3 Scree Test

```
scree <- fa.parallel(PC_R[, c(1:20)], fa = "pc")
```



```
## Parallel analysis suggests that the number of factors =  NA  and the number of components =  11
```

The KMO and scree tests suggest no evidence for strong principal components in the data.

3.4.4 PCA

```
PCA_R <- principal(R, nfactors = 1, rotate = "none", n.obs = 500,
    residuals = TRUE)
PCA_R

## Principal Components Analysis
## Call: principal(r = R, nfactors = 1, residuals = TRUE, rotate = "none",
##     n.obs = 500)
## Standardized loadings (pattern matrix) based upon correlation matrix
##       PC1      h2   u2 com
## v1   0.30 0.08743 0.91   1
## v2   0.20 0.04173 0.96   1
## v3   0.20 0.03938 0.96   1
## v4   0.38 0.14082 0.86   1
## v5  -0.34 0.11466 0.89   1
## v6   0.12 0.01432 0.99   1
## v7   0.20 0.03948 0.96   1
## v8   0.28 0.07744 0.92   1
## v9  -0.26 0.06615 0.93   1
## v10 -0.06 0.00401 1.00   1
## v11  0.45 0.20398 0.80   1
## v12 -0.14 0.01950 0.98   1
## v13  0.39 0.14884 0.85   1
## v14 -0.32 0.10222 0.90   1
## v15  0.03 0.00091 1.00   1
## v16 -0.11 0.01159 0.99   1
## v17 -0.03 0.00096 1.00   1
## v18  0.51 0.25797 0.74   1
## v19 -0.13 0.01789 0.98   1
## v20 -0.30 0.08835 0.91   1
##
##                 PC1
## SS loadings    1.48
## Proportion Var 0.07
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.07
##  with the empirical chi square  924.7  with prob <  4.2e-104
##
## Fit based upon off diagonal values = -0.44
```

The original principal components analysis would have indicated that a single component, and a single t-test, would have provided an adequate test of group differences. When the correct matrix is analyzed (residuals), there is no evidence for multidimensionality, indicating the need for 20 individual t-tests (and perhaps appropriate Type I error protection).

4 Multicollinearity

Principal components analysis can also be used to solve multicollinearity problems.

4.1 Create a Multicollinear Data Set

Here we create a multivariate normal data set (N=100). The means (equal to 0) are defined in the vector, mu. The variance-covariance matrix is defined by the matrix, sigma. Because we will use standard normal variables, the covariance matrix is just a correlation matrix. The predictors (first 6 variables) are intercorrelated in the population

at .9. Each predictor has a correlation with the DV (last variable) of .6

$$R = \begin{bmatrix} 1.00 & 0.90 & 0.90 & 0.90 & 0.90 & 0.90 & 0.60 \\ 0.90 & 1.00 & 0.90 & 0.90 & 0.90 & 0.90 & 0.60 \\ 0.90 & 0.90 & 1.00 & 0.90 & 0.90 & 0.90 & 0.60 \\ 0.90 & 0.90 & 0.90 & 1.00 & 0.90 & 0.90 & 0.60 \\ 0.90 & 0.90 & 0.90 & 0.90 & 1.00 & 0.90 & 0.60 \\ 0.90 & 0.90 & 0.90 & 0.90 & 0.90 & 1.00 & 0.60 \\ 0.60 & 0.60 & 0.60 & 0.60 & 0.60 & 0.60 & 1.00 \end{bmatrix}$$

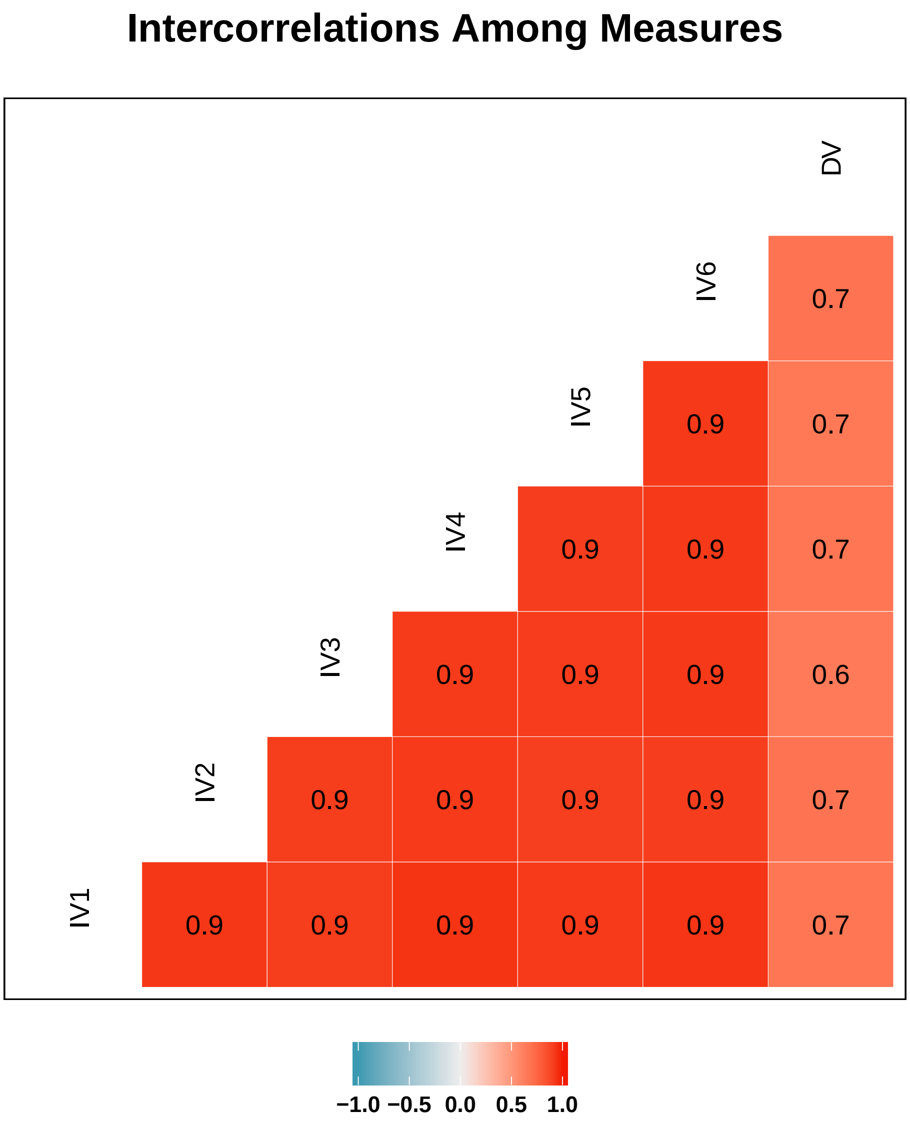
```
mu <- matrix(c(0, 0, 0, 0, 0, 0, 0), nrow = 7, ncol = 1)
sigma <- matrix(c(1, 0.9, 0.9, 0.9, 0.9, 0.9, 0.6, 0.9, 1, 0.9, 0.9,
    0.9, 0.9, 0.6, 0.9, 0.9, 1, 0.9, 0.9, 0.9, 0.6, 0.9, 0.9, 0.9,
    1, 0.9, 0.9, 0.6, 0.9, 0.9, 0.9, 0.9, 1, 0.9, 0.6, 0.9, 0.9, 0.9,
    0.9, 0.9, 1, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 0.6, 1), nrow = 7,
    ncol = 7, byrow = TRUE)
Data <- data.frame(mvrnorm(n = 100, mu, sigma, tol = 0.000001, empirical = FALSE))
names(Data) <- c("IV1", "IV2", "IV3", "IV4", "IV5", "IV6", "DV")
```

4.1.1 Descriptive Statistics

```
describe(Data)

##     vars   n  mean   sd median trimmed  mad   min  max range
## IV1    1 100 -0.05 1.11   0.03   -0.04 1.00 -2.78 2.65  5.44
## IV2    2 100 -0.05 1.12  -0.06   -0.03 1.12 -2.74 2.31  5.05
## IV3    3 100  0.00 1.09   0.04   -0.01 1.14 -3.08 2.96  6.04
## IV4    4 100 -0.04 1.11   0.04   -0.03 1.17 -2.49 2.15  4.64
## IV5    5 100 -0.02 1.10  -0.01   -0.02 1.10 -2.28 2.39  4.68
## IV6    6 100  0.06 1.08   0.03    0.05 1.07 -2.64 2.42  5.07
## DV     7 100  0.06 1.02   0.03    0.05 1.03 -2.45 2.51  4.97
##      skew kurtosis   se
## IV1 -0.12    -0.44 0.11
## IV2 -0.14    -0.30 0.11
## IV3  0.04     0.06 0.11
## IV4 -0.06    -0.72 0.11
## IV5  0.01    -0.66 0.11
## IV6  0.00    -0.50 0.11
## DV   0.07    -0.21 0.10

R <- cor(Data)
round(R, 2)

##      IV1  IV2  IV3  IV4  IV5  IV6   DV
## IV1 1.00 0.93 0.91 0.94 0.92 0.94 0.67
## IV2 0.93 1.00 0.91 0.92 0.91 0.91 0.68
## IV3 0.91 0.91 1.00 0.92 0.91 0.92 0.65
## IV4 0.94 0.92 0.92 1.00 0.91 0.92 0.67
## IV5 0.92 0.91 0.91 0.91 1.00 0.93 0.65
## IV6 0.94 0.91 0.92 0.92 0.93 1.00 0.68
## DV  0.67 0.68 0.65 0.67 0.65 0.68 1.00
```

A heat map for the correlation matrix suggests no clear patterning in the data.

```
ggcorr(Data[, c(1:7)], label = TRUE, angle = 90, hjust = 0.1, size = 4,
    digits = 2) + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Intercorrelations Among Measures")
```



Intercorrelations Among Measures

4.2 Linear Model with Multicollinear Predictors

The 6 multicollinear predictors are used to account for variance in the DV. The overall proportion of variance accounted for is substantial, but because of the multicollinearity, just one of the predictors is individually significant (Type III sums of squares) despite the fact that each independent variable has a substantial (and in the population, equal) relationship with the outcome.

4.2.1 Regression Summary

```
# Note that the following provides Type III sums of squares.
lm_fit_1 <- lm(DV ~ IV1 + IV2 + IV3 + IV4 + IV5 + IV6, data = Data)
summary(lm_fit_1)
```

```
##
## Call:
## lm(formula = DV ~ IV1 + IV2 + IV3 + IV4 + IV5 + IV6, data = Data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.877 -0.468  0.030  0.541  1.566
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0558     0.0787    0.71     0.48
## IV1          -0.0229     0.2607   -0.09     0.93
## IV2           0.2914     0.2130    1.37     0.17
## IV3          -0.1137     0.2196   -0.52     0.61
## IV4           0.1202     0.2368    0.51     0.61
## IV5           0.0129     0.2134    0.06     0.95
## IV6           0.3738     0.2431    1.54     0.13
##
## Residual standard error: 0.752 on 93 degrees of freedom
## Multiple R-squared:  0.488,Adjusted R-squared:  0.455
## F-statistic: 14.8 on 6 and 93 DF,  p-value: 8.37e-12
```

4.2.2 Variance Inflation Factor

A useful way to assess multicollinearity is by the variance inflation factor (VIF). The VIF for a variable increases from 1 to the extent that the variable is highly related to the other predictors. VIF is defined as follows:

$$VIF_j = \frac{1}{1 - R_j^2}$$

in which R_j^2 is the squared multiple correlation from regressing on predictor on the remaining predictors. The VIF indicates by how much the variance of a regression coefficient is multiplied because of collinearity.

The reciprocal of the VIF is the tolerance, which is the amount of variance in a predictor that is unique from the other predictors.

```
vif(lm_fit_1)

##    IV1    IV2    IV3    IV4    IV5    IV6
## 14.554  9.939 10.010 12.062  9.688 12.002

1/vif(lm_fit_1)

##     IV1     IV2     IV3     IV4     IV5     IV6
## 0.06871 0.10062 0.09990 0.08290 0.10322 0.08332

lm_fit_2 <- lm(IV1 ~ IV2 + IV3 + IV4 + IV5 + IV6, data = Data)
summary(lm_fit_2)

##
## Call:
```

```
## lm(formula = IV1 ~ IV2 + IV3 + IV4 + IV5 + IV6, data = Data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5821 -0.1696  0.0119  0.2047  0.7535
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.0421     0.0308   -1.37  0.17504
## IV2           0.2483     0.0803    3.09  0.00261
## IV3          -0.0190     0.0869   -0.22  0.82754
## IV4           0.3146     0.0879    3.58  0.00055
## IV5           0.1567     0.0829    1.89  0.06167
## IV6           0.2990     0.0911    3.28  0.00145
##
## Residual standard error: 0.297 on 94 degrees of freedom
## Multiple R-squared:  0.931,Adjusted R-squared:  0.928
## F-statistic:  255 on 5 and 94 DF,  p-value: <2e-16
```

The VIF values are all very high, indicating serious multicollinearity.

4.3 PCA of the Multicollinear Predictors

A principal components analysis indicates just a single underlying dimension for the predictors, accounting for over 90% of the variance in the original data.. The principal component score is retained for further analyses.

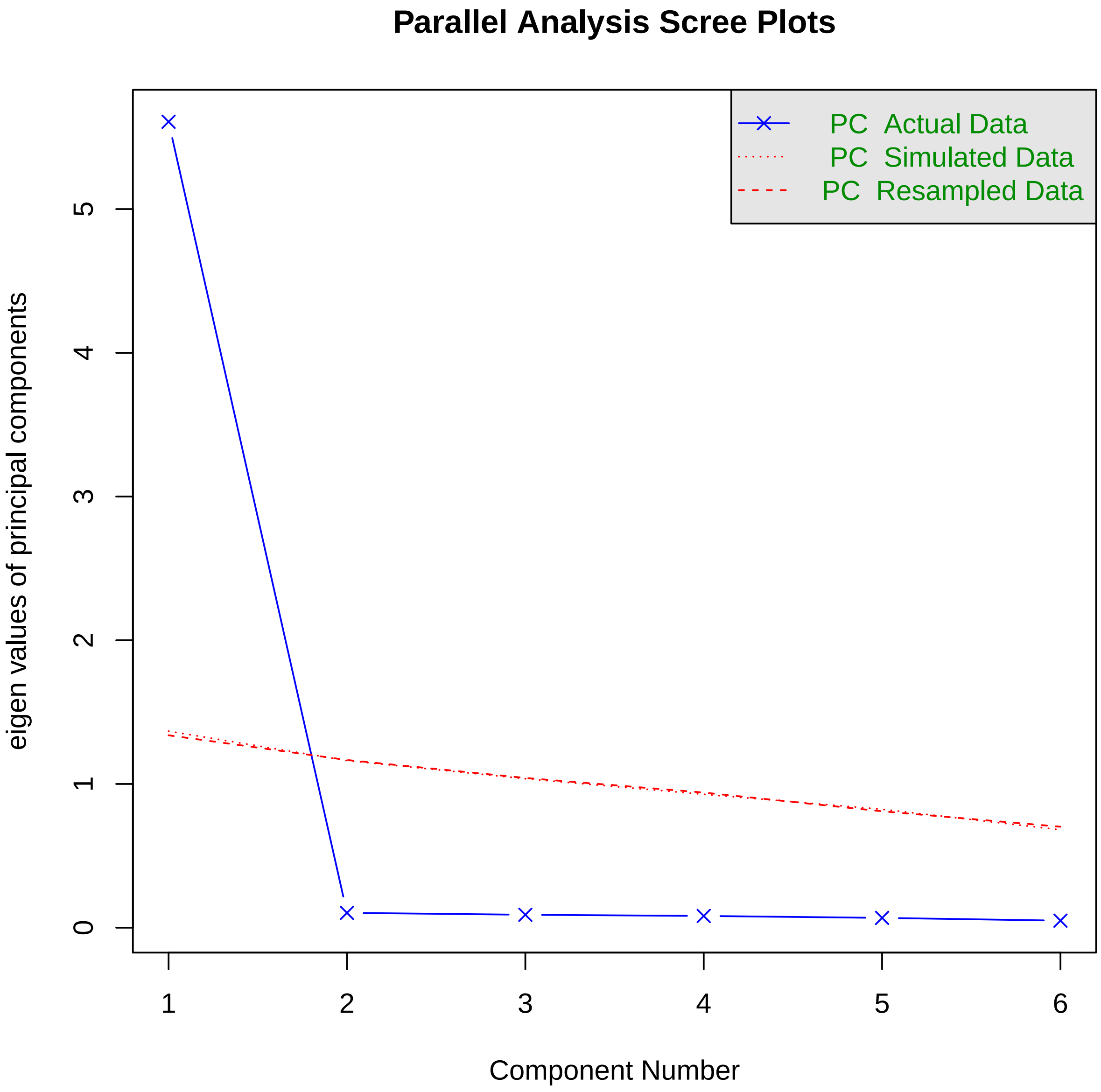
4.3.1 Should a PCA be Conducted?

```
KMO(R)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = R)
## Overall MSA =  0.95
## MSA for each item =
##  IV1  IV2  IV3  IV4  IV5  IV6   DV
## 0.93 0.95 0.95 0.95 0.96 0.94 0.98
```

```
cortest.bartlett(R = R, n = 500)

## $chisq
## [1] 5865
##
## $p.value
## [1] 0
##
## $df
## [1] 21
```

4.3.2 Scree Test

```
scree <- fa.parallel(Data[, c(1:6)], fa = "pc")
```



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

4.3.3 PCA Summary

```
PCA_3 <- principal(Data[, 1:6], nfactors = 1, rotate = "none", n.obs = 100,
    residuals = TRUE)
PCA_3

## Principal Components Analysis
## Call: principal(r = Data[, 1:6], nfactors = 1, residuals = TRUE, rotate = "none",
##     n.obs = 100)
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1   h2    u2 com
## IV1 0.97 0.95 0.052   1
## IV2 0.96 0.93 0.074   1
## IV3 0.96 0.93 0.072   1
## IV4 0.97 0.94 0.060   1
## IV5 0.96 0.93 0.074   1
## IV6 0.97 0.94 0.061   1
##
##                 PC1
## SS loadings    5.61
## Proportion Var 0.93
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.01
##  with the empirical chi square  0.64  with prob <  1
##
## Fit based upon off diagonal values = 1

Data <- cbind(Data, PCA_3$scores)
```

4.4 Linear Model with Principal Component Score

The outcome variable is predicted from the single principal component score, with little loss of information.

```
summary(lm(DV ~ PC1, data = Data))

##
## Call:
## lm(formula = DV ~ PC1, data = Data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8716 -0.4799  0.0677  0.5921  1.7136
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0597     0.0742    0.80     0.42
## PC1           0.7028     0.0745    9.43  2.1e-15
##
## Residual standard error: 0.742 on 98 degrees of freedom
## Multiple R-squared:  0.476,Adjusted R-squared:  0.47
## F-statistic: 88.9 on 1 and 98 DF,  p-value: 2.11e-15
```

5 Ordinal Data

What about data that are strictly ordinal but often treated as though continuous? The Need for Cognition Scale used in our first PCA example had a 5-point rating scale:

1 = very characteristic of me
2 = somewhat characteristic of me
3 = neutral
4 = somewhat uncharacteristic of me
5 = very uncharacteristic of me

If the underlying construct is viewed as continuous, then crude categories such as this will attenuate correlations.

Does it matter that we are assuming the variables to be continuous when they are only approximately so? We can investigate this question by converting the empirical correlations to their expected values for the underlying (and truly) continuous and bivariate normally distributed latent variables. These are called polychoric correlations. Then we can repeat the principal components analysis on the polychoric correlations and compare the results to the original analyses.

5.1 Need for Cognition Data

```
# Get the data from the working directory.
setwd("C:\\Courses\\Psychology 516\\PowerPoint\\2018")
NC <- read.table("need_for_cognition.csv", sep = ",", header = TRUE)
NC <- as.data.frame(NC)
NC <- na.omit(NC)


# Reverse score items
NC$item_1 <- 6 - NC$item_1
NC$item_2 <- 6 - NC$item_2
NC$item_6 <- 6 - NC$item_6
NC$item_10 <- 6 - NC$item_10
NC$item_11 <- 6 - NC$item_11
NC$item_13 <- 6 - NC$item_13
NC$item_14 <- 6 - NC$item_14
NC$item_15 <- 6 - NC$item_15
NC$item_18 <- 6 - NC$item_18
```

5.2 Conversion to Ordered Factors

We need to convert the items to ordered factors so they are treated correctly by the hetcor() function. Otherwise they will be treated as continuous and hetcor() will just produce Pearson product-moment correlations.

```
NC$I1 <- ordered(NC$item_1, levels = c(1, 2, 3, 4, 5))
NC$I2 <- ordered(NC$item_2, levels = c(1, 2, 3, 4, 5))
NC$I3 <- ordered(NC$item_3, levels = c(1, 2, 3, 4, 5))
NC$I4 <- ordered(NC$item_4, levels = c(1, 2, 3, 4, 5))
NC$I5 <- ordered(NC$item_5, levels = c(1, 2, 3, 4, 5))
NC$I6 <- ordered(NC$item_6, levels = c(1, 2, 3, 4, 5))
NC$I7 <- ordered(NC$item_7, levels = c(1, 2, 3, 4, 5))
NC$I8 <- ordered(NC$item_8, levels = c(1, 2, 3, 4, 5))
NC$I9 <- ordered(NC$item_9, levels = c(1, 2, 3, 4, 5))
NC$I10 <- ordered(NC$item_10, levels = c(1, 2, 3, 4, 5))
NC$I11 <- ordered(NC$item_11, levels = c(1, 2, 3, 4, 5))
NC$I12 <- ordered(NC$item_12, levels = c(1, 2, 3, 4, 5))
NC$I13 <- ordered(NC$item_13, levels = c(1, 2, 3, 4, 5))
NC$I14 <- ordered(NC$item_14, levels = c(1, 2, 3, 4, 5))
NC$I15 <- ordered(NC$item_15, levels = c(1, 2, 3, 4, 5))
NC$I16 <- ordered(NC$item_16, levels = c(1, 2, 3, 4, 5))
NC$I17 <- ordered(NC$item_17, levels = c(1, 2, 3, 4, 5))
NC$I18 <- ordered(NC$item_18, levels = c(1, 2, 3, 4, 5))
```

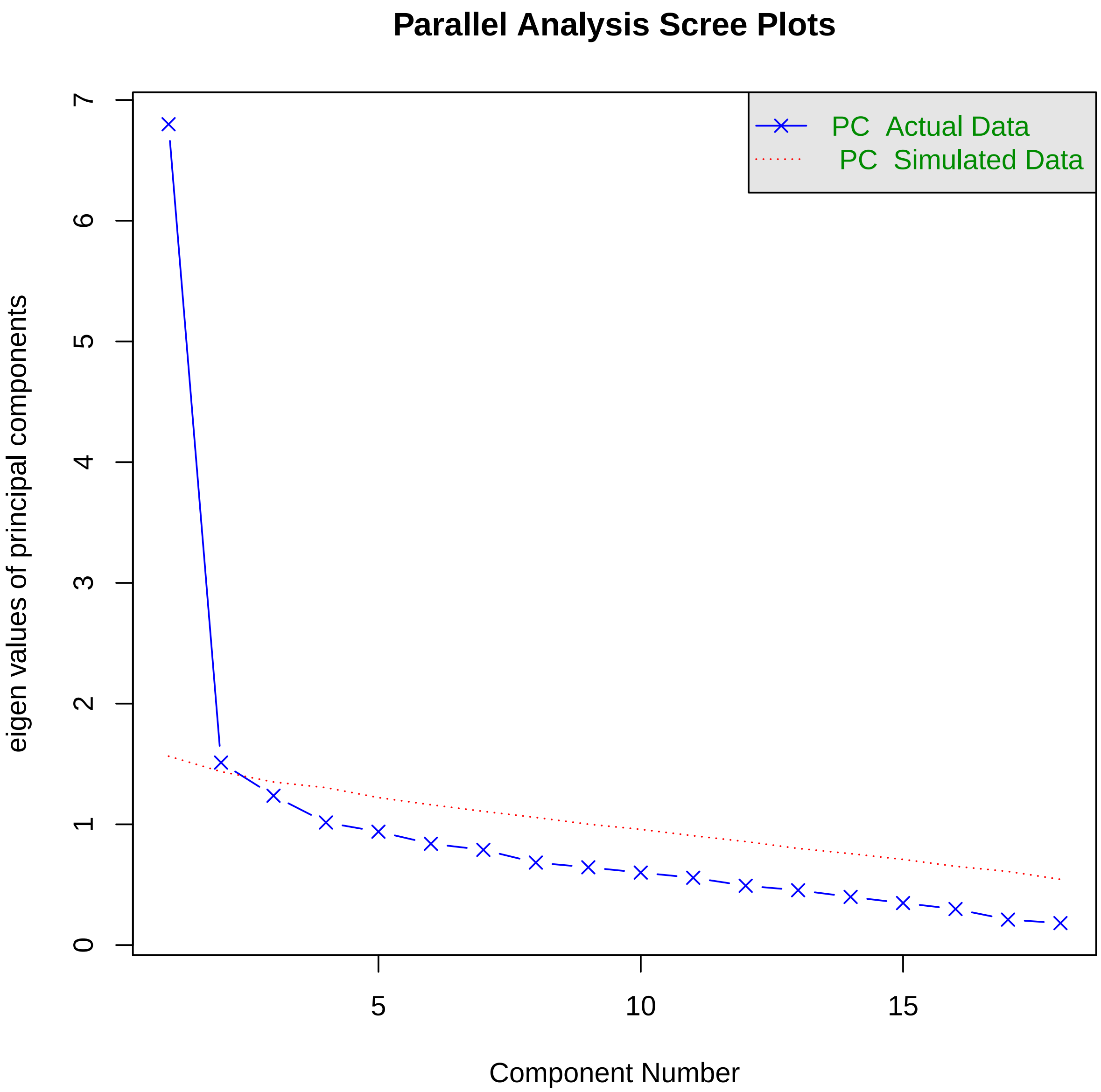
5.3 Polychoric Correlations

The hetcor() function produces the expected correlations under the assumption that the underlying latent variables are continuous, using maximum likelihood.

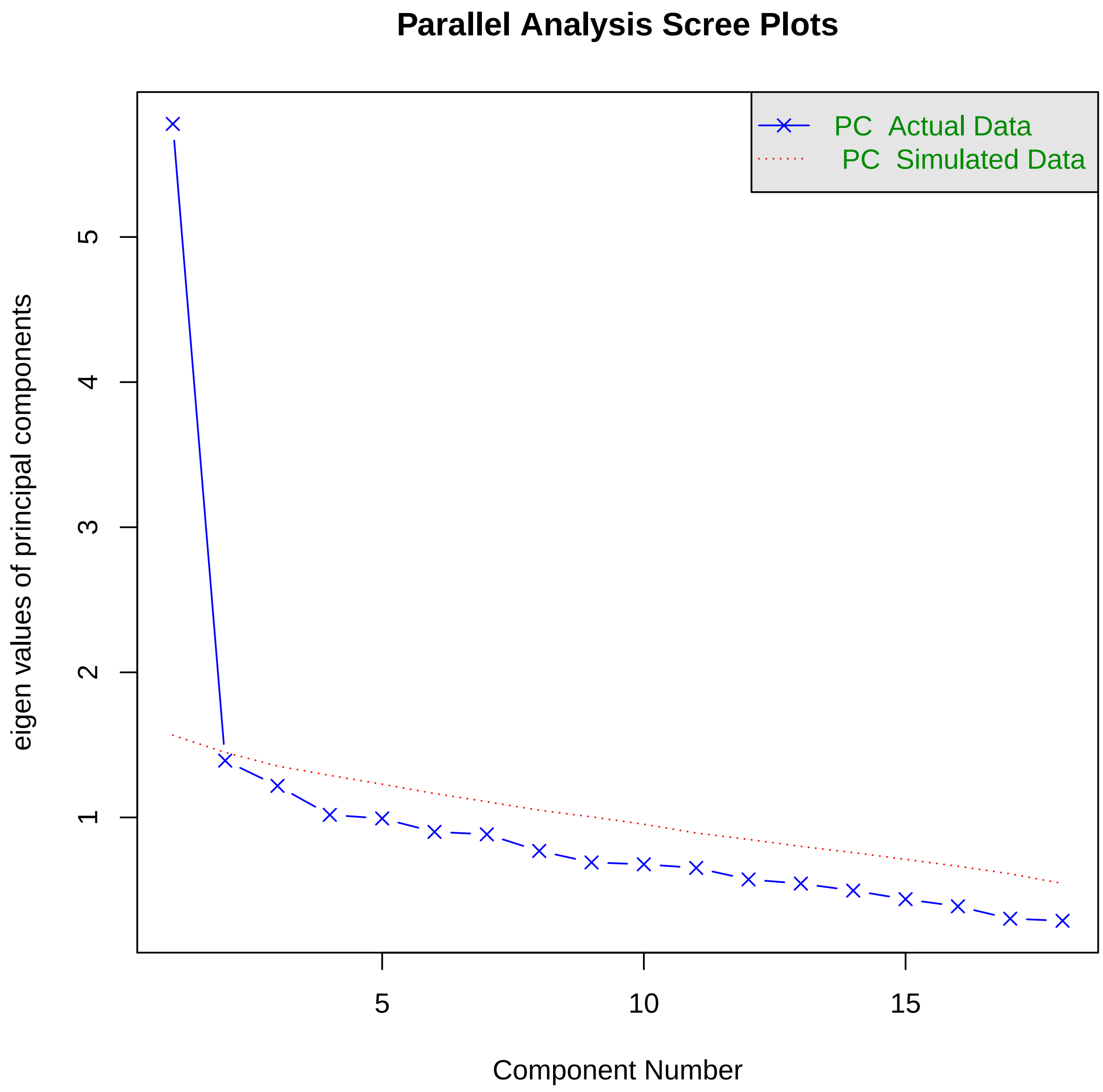
```
PR <- hetcor(NC[, 19:36], ML = TRUE, pd = TRUE)$correlations
R <- cor(NC[, 1:18])

# Mean difference between the off-diagonals of the two matrices.
PR_2 <- PR
diag(PR_2) <- NA
R_2 <- R
diag(R_2) <- NA
mean(PR_2 - R_2, na.rm = TRUE)

## [1] 0.05664
```

5.4 Scree Tests

```
scree_PR <- fa.parallel(PR, n.obs = 195, fa = "pc")
```

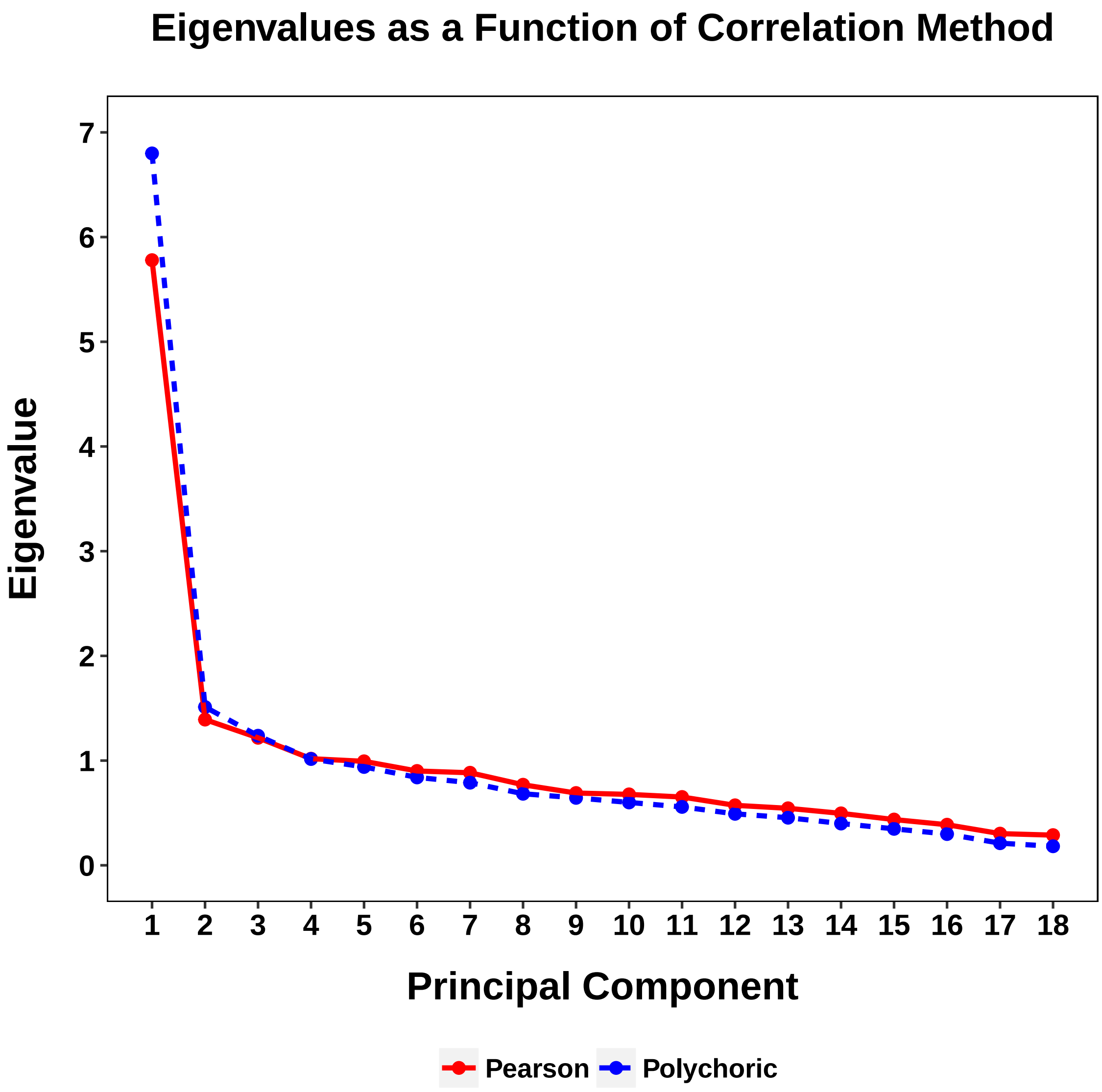


```
## Parallel analysis suggests that the number of factors =  NA  and the number of components =  2
scree_R <- fa.parallel(R, n.obs = 195, fa = "pc")
```



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

spr <- scree_PR$pc.values
sr <- scree_R$pc.values
scree_total <- matrix(c(sr, spr))
scree_total <- as.data.frame(scree_total)
names(scree_total) <- c("eigenvalues")
scree_total$component <- c(seq(1, 18), seq(1, 18))
scree_total$method <- c(rep(1, 18), rep(2, 18))
scree_total$method_F <- factor(scree_total$method, levels = c(1, 2),
    labels = c("Pearson", "Polychoric"))
```

```r
ggplot(scree_total, aes(x = component, y = eigenvalues, color = method_F)) +
    geom_point(size = 2) + geom_line(aes(linetype = method_F), size = 1) +
    scale_color_manual(values = c("red", "blue")) + coord_cartesian(xlim = c(1,
    18), ylim = c(0, 7)) + scale_x_continuous(breaks = c(seq(1, 18,
    1))) + scale_y_continuous(breaks = seq(0, 7, 1)) + xlab("Principal Component") +
    ylab("Eigenvalue") + theme(text = element_text(size = 14, family = "sans",
    color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
    plot.title = element_text(size = 16, face = "bold", margin = margin(0,
        0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
        linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + ggtitle("Eigenvalues as a Function of Correlation Method")
```



5.5 PCA

```
PCA_PR <- principal(PR, nfactors = 1, rotate = "none", n.obs = 195,
    residuals = TRUE)
PCA_R <- principal(R, nfactors = 1, rotate = "none", n.obs = 195,
    residuals = TRUE)

PCA_PR

## Principal Components Analysis
## Call: principal(r = PR, nfactors = 1, residuals = TRUE, rotate = "none",
##     n.obs = 195)
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1    h2   u2 com
## I1  0.64 0.405 0.59   1
## I2  0.78 0.610 0.39   1
## I3  0.72 0.512 0.49   1
## I4  0.71 0.506 0.49   1
## I5  0.75 0.569 0.43   1
## I6  0.47 0.224 0.78   1
## I7  0.60 0.355 0.64   1
## I8  0.51 0.257 0.74   1
## I9  0.59 0.350 0.65   1
## I10 0.68 0.466 0.53   1
## I11 0.76 0.571 0.43   1
## I12 0.70 0.490 0.51   1
## I13 0.60 0.354 0.65   1
## I14 0.58 0.341 0.66   1
## I15 0.49 0.239 0.76   1
## I16 0.43 0.182 0.82   1
## I17 0.57 0.321 0.68   1
## I18 0.22 0.048 0.95   1
##
##                  PC1
## SS loadings    6.80
## Proportion Var 0.38
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.08
##  with the empirical chi square  417.1  with prob <  1.5e-30
##
## Fit based upon off diagonal values = 0.94
```

```
PCA_R
```

```
## Principal Components Analysis
## Call: principal(r = R, nfactors = 1, residuals = TRUE, rotate = "none",
##     n.obs = 195)
## Standardized loadings (pattern matrix) based upon correlation matrix
##          PC1    h2   u2 com
## item_1  0.60 0.364 0.64   1
## item_2  0.74 0.553 0.45   1
## item_3  0.61 0.370 0.63   1
## item_4  0.66 0.436 0.56   1
## item_5  0.70 0.483 0.52   1
## item_6  0.44 0.194 0.81   1
## item_7  0.55 0.297 0.70   1
## item_8  0.49 0.245 0.76   1
## item_9  0.56 0.317 0.68   1
## item_10 0.62 0.387 0.61   1
## item_11 0.68 0.459 0.54   1
## item_12 0.62 0.388 0.61   1
## item_13 0.56 0.312 0.69   1
## item_14 0.55 0.305 0.69   1
## item_15 0.41 0.164 0.84   1
## item_16 0.39 0.156 0.84   1
## item_17 0.55 0.304 0.70   1
## item_18 0.21 0.044 0.96   1
##
##                 PC1
## SS loadings    5.78
## Proportion Var 0.32
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.08
##  with the empirical chi square  350.4  with prob <  4.5e-21
##
## Fit based upon off diagonal values = 0.93
```

As expected, the principal components analysis results are clearer when based on the polychoric correlations. Nonetheless, the differences are not great and conclusions would not change, in this example.