The research objective is to obtain a general measure of issue attitudes, using a summated rating scale. The “subjects” in this analysis are 320 respondents to a public opinion survey (the 1988 CPS National Election Study). These people were asked their own positions on each of seven issues. For each issue, the response format was identical. There were seven ordinal categories, and scores were assigned to the categories using successive integers from 1 to 7. For every variable, higher scores indicate more conservative issue positions.

Key to Variables:

- gsr: Respondent’s attitude toward government spending
- dsr: Respondent’s attitude toward defense spending
- gir: Respondent’s attitude toward government-financed health insurance
- jsr: Respondent’s attitude toward government-guaranteed jobs
- mar: Respondent’s attitude toward government aid to minorities
- rsr: Respondent’s attitude toward negotiations with the soviet union
- wrr: Respondent’s attitude toward women's rights

Output: Log of R session, showing commands to read data, calculate reliability, create summated rating scale, obtain summary statistics, and perform item analysis.
> ### Create a subset of the original dataset, that includes only the issue questions.
> issues <- data88[, c("gsr", "dsr", "gir", "jsr", "mar", "rsr", "wrr")]
> issues[1:5, ]
gsr dsr gir jsr mar rsr wrr
1 2 6 2 2 5 2 5
2 2 5 1 1 7 1 1
3 2 2 2 3 2 1 3
4 7 7 1 1 7 1 1
5 5 3 6 4 4 2 6
> ### Correlation matrix for issue variables
> cor(issues)
gsr dsr gir jsr mar rsr wrr
gsr 1.0000000 0.1929926 0.38883082 0.32989001 0.09021492 0.19531527 0.3480497
dsr 0.19299263 1.0000000 0.18574717 0.26818867 0.34000801 0.12573725 0.2664576
gir 0.38883082 0.1857472 1.00000000 0.42955866 0.08177226 0.06058244 0.3886204
jsr 0.32989001 0.2681887 0.42955866 1.00000000 0.03276570 0.05663687 0.4810982
mar 0.09021492 0.3400080 0.08177226 0.03276570 1.00000000 0.14141280 0.1286260
rsr 0.19531527 0.1257372 0.06058244 0.05663687 0.14141280 1.00000000 0.0774502
wrr 0.34804967 0.2664576 0.38862045 0.48109820 0.12862596 0.07745020 1.0000000
> ### Calculate mean bivariate correlation
> n.vars <- nrow(cor(issues))
> n.vars
[1] 7
> sum.corr <- sum(cor(issues)) - n.vars
> sum.corr
[1] 9.219911
> mean.corr <- sum.corr / (n.vars * (n.vars - 1))
> mean.corr
[1] 0.2195217
> ### Calculate alpha for seven-item scale from formula
> my.alpha <- (n.vars * mean.corr) /
+                (1 + (mean.corr * (n.vars - 1)))
> my.alpha
[1] 0.6631703
> ### Standardize issue variables before using the R function to calculate alpha
> issues2 <- as.data.frame(scale(issues))
> mean(issues2)
gsr dsr gir jsr mar
-1.006140e-16 1.245924e-16 3.159704e-17 9.435020e-17 -3.356385e-17
rsr
-4.027404e-17 1.945198e-16
> sd(issues2)
### Use "alpha" function to create an object containing parts of a reliability analysis.

```r
reliab.issues <- alpha(issues2)
names(reliab.issues)
[1] "total"       "alpha.drop"  "item.stats"  "response.freq"
[5] "keys"        "scores"      "nvar"         "boot.ci"
[9] "boot"        "call"        "title"
```

### Print the alpha coefficient, itself

```r
reliab.issues$total
raw_alpha std.alpha   G6(smc) average_r      S/N        ase
0.6631703 0.6631703 0.6638371 0.2195217 1.968859 0.03920954
mean        sd
3.934e-17 0.5753422
```

### Use "apply" function to create summated rating scale of issue responses, defined as the mean of each person's responses across the seven items

```r
scale <- apply(issues, MARGIN = 1, FUN = mean)
```

### Summary statistics on the scale

```r
summary(scale)
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.  
1.000   3.000   3.857   3.754   4.429   6.857 
```

### The following function produces Figure 1

```r
histogram(~scale, aspect = .75, xlab = "Seven-item summed rating scale")
```

### Perform an item analysis on the scale

```r
reliab.issues$alpha.drop
raw_alpha std.alpha   G6(smc) average_r      S/N        ase
gsr 0.6063973 0.6063973 0.6049468 0.2043108 1.540633 0.04575208
ds 0.6222273 0.6222273 0.6162349 0.2153883 1.647095 0.04451838
gir 0.6073922 0.6073922 0.6010314 0.2049896 1.547071 0.04577221
jsr 0.6011794 0.6011794 0.5878788 0.2007878 1.507393 0.04629142
mar 0.6702108 0.6702108 0.6543194 0.2530104 2.032240 0.04629142
rsr 0.6822245 0.6822245 0.6731722 0.2635214 2.146876 0.04010658
wrr 0.5918563 0.5918563 0.5864522 0.1946436 1.450117 0.04692616
### Calculate restscores and plot against corresponding items

```r
restscores <- matrix(nrow = nrow(issues), ncol = ncol(issues))

names <- rep("rest", times = ncol(issues))

for (j in 1:ncol(issues)) {
  restscores[, j] <- apply(issues[, -j], MARGIN = 1, FUN = mean)
  names[j] <- paste(names[j], colnames(issues)[j], sep = ".")
}

colnames(restscores) <- names

### The following functions produce the graphs shown in Figure 2

set.seed(1357)
xyplot(issues[,1] ~ restscores[,1],
  aspect = 1,
  panel = function (x, y) {
    panel.xyplot(jitter(x), jitter(y), col = "gray")
    panel.loess(x, y, span = .7, col = "black",
      family = "symmetric", degree = 2)
  },
  main = "A. Plot for variable \"gsr\"")

set.seed(1357)
xyplot(issues[,2] ~ restscores[,2],
  aspect = 1,
  panel = function (x, y) {
    panel.xyplot(jitter(x), jitter(y), col = "gray")
    panel.loess(x, y, span = .7, col = "black",
      family = "symmetric", degree = 2)
  },
  main = "B. Plot for variable \"dsr\"")

set.seed(1357)
xyplot(issues[,3] ~ restscores[,3],
  aspect = 1,
  panel = function (x, y) {
    panel.xyplot(jitter(x), jitter(y), col = "gray")
    panel.loess(x, y, span = .7, col = "black",
      family = "symmetric", degree = 2)
  },
  main = "C. Plot for variable \"gir\"")

set.seed(1357)
xyplot(issues[,4] ~ restscores[,4],
  aspect = 1,
  panel = function (x, y) {
    panel.xyplot(jitter(x), jitter(y), col = "gray")
    panel.loess(x, y, span = .7, col = "black",
      family = "symmetric", degree = 2)
  },
  main = "D. Plot for variable \"jsr\"")
```

> set.seed(1357)
> xyplot(issues[,5] ~ restscores[,5],
+     aspect = 1,
+     panel = function (x, y) {
+       panel.xyplot(jitter(x), jitter(y), col = "gray")
+       panel.loess(x, y, span = .7, col = "black",
+         family = "symmetric", degree = 2)
+     },
+     main = "E. Plot for variable \"mar\"")
> set.seed(1357)
> xyplot(issues[,6] ~ restscores[,6],
+     aspect = 1,
+     panel = function (x, y) {
+       panel.xyplot(jitter(x), jitter(y), col = "gray")
+       panel.loess(x, y, span = .7, col = "black",
+         family = "symmetric", degree = 2)
+     },
+     main = "F. Plot for variable \"rsr\"")
> set.seed(1357)
> xyplot(issues[,7] ~ restscores[,7],
+     aspect = 1,
+     panel = function (x, y) {
+       panel.xyplot(jitter(x), jitter(y), col = "gray")
+       panel.loess(x, y, span = .7, col = "black",
+         family = "symmetric", degree = 2)
+     },
+     main = "G. Plot for variable \"wrr\"")
>

**Figure 1:** Histogram of summated rating scale scores.
Figure 2: Graphs showing each item plotted versus the corresponding restscore. A loess curve is superimposed over the data in each case.

A. Plot for variable "gsr"

B. Plot for variable "dsr"

C. Plot for variable "gir"

D. Plot for variable "jsr"
Figure 2: Graphs showing each item plotted versus the corresponding restscore. A loess curve is superimposed over the data in each case (Continued).

E. Plot for variable "mar"

F. Plot for variable "rsr"

G. Plot for variable "wrr"