What statistical analysis should I use in R?

Authored by

stats writer

June 28, 2024

The appropriate method for rewriting a sentence is to carefully analyze the original sentence and determine the main idea or message it is conveying. Then, using proper grammar and sentence structure, rephrase the sentence in a clear and concise manner without changing the intended meaning. It is important to also consider the tone and style of writing when rewriting a sentence. Overall, the proper way to rewrite a sentence is to effectively communicate the same information in a more effective and polished manner.

What statistical analysis should I use? Statistical analyses using R

Version info: Code for this page was tested in R 2.15.2.

Introduction

This page shows how to perform a number of statistical tests using R.
Each section gives a brief description of the aim of the statistical test,
when it is used, an example showing the R commands and R output with
a brief interpretation of the output. You can see the page

Choosing the Correct Statistical Test for a table that shows an overview
of when each test is appropriate to use. In deciding which test is
appropriate to use, it is important to consider the type

of variables that you have (i.e., whether your variables are categorical, ordinal or interval and whether they are normally distributed), see

What is the difference between categorical, ordinal and interval variables?

for more information on this.

Setup

```
hsb2 <-
within(read.csv("https://stats.idre.ucla.edu/stat/data/hsb2.csv"), {
race <- as.factor(race)
schtyp <- as.factor(schtyp)
prog <- as.factor(prog)
})

attach(hsb2)
```

One sample t-test

```
t.test(write, mu = 50)

##
```

```
## One Sample t-test
##
## data: write
## t = 4.14, df = 199, p-value = 5.121e-05
## alternative hypothesis: true mean is not equal to 50
## 95 percent confidence interval:
## 51.5 54.1
## sample estimates:
## mean of x
## 52.8
```

One sample median test

```
wilcox.test(write, mu = 50)
```

```
##
## Wilcoxon signed rank test with continuity correction
##
## data: write
## V = 13177, p-value = 3.702e-05
## alternative hypothesis: true location is not equal to 50
```

Binomial test

```
prop.test(sum(female), length(female), p = 0.5)

##
## 1-sample proportions test with continuity correction
##
## data: sum(female) out of length(female), null probability 0.5
## X-squared = 1.45, df = 1, p-value = 0.2293
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
## 0.473 0.615
## sample estimates:
## p
## 0.545
```

Chi-square goodness of fit

```
chisq.test(table(race), p = c(10, 10, 10, 70)/100)

##
## Chi-squared test for given probabilities
##
## data: table(race)
## X-squared = 5.03, df = 3, p-value = 0.1697
```

Two independent samples t-test

```
t.test(write ~ female)

##
## Welch Two Sample t-test
##
## data: write by female
## t = -3.66, df = 170, p-value = 0.0003409
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -7.50 -2.24
## sample estimates:
## mean in group 0 mean in group 1
## 50.1 55.0
```

Wilcoxon-Mann-Whitney test

```
wilcox.test(write ~ female)

##
## Wilcoxon rank sum test with continuity correction
##
## data: write by female
```

```
## W = 3606, p-value = 0.0008749
## alternative hypothesis: true location shift is not equal to 0
```

Chi-square test

```
chisq.test(table(female, schtyp))
```

```
##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table(female, schtyp)
## X-squared = 5e-04, df = 1, p-value = 0.9815
```

Fisher's exact test

```
fisher.test(table(race, schtyp))
```

```
##
## Fisher's Exact Test for Count Data
##
## data: table(race, schtyp)
## p-value = 0.5975
```

alternative hypothesis: two.sided

One-way ANOVA

summary(aov(write ~ prog))

Df Sum Sq Mean Sq F value Pr(>F)
prog 2 3176 1588 21.3 4.3e-09 ***
Residuals 197 14703 75

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Kruskal Wallis test

kruskal.test(write, prog)

##
Kruskal-Wallis rank sum test
##
data: write and prog
Kruskal-Wallis chi-squared = 34, df = 2, p-value = 4.047e-08

Paired t-test

```
t.test(write, read, paired = TRUE)

##
## Paired t-test
##
## data: write and read
## t = 0.867, df = 199, p-value = 0.3868
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.694 1.784
## sample estimates:
## mean of the differences
## 0.545
```

Wilcoxon signed rank sum test

```
wilcox.test(write, read, paired = TRUE)

##
## Wilcoxon signed rank test with continuity correction
##
## data: write and read
## V = 9261, p-value = 0.3666
```

alternative hypothesis: true location shift is not equal to 0

McNemar test

X <- matrix(c(172, 7, 6, 15), 2, 2)
mcnemar.test(X)

##
McNemar's Chi-squared test with continuity correction
##
data: X
McNemar's chi-squared = 0, df = 1, p-value = 1

One-way repeated measures ANOVA

require(car)

Loading required package: car

Loading required package: MASS

Loading required package: nnet

```
## Loading required package: survival

## Loading required package: splines

require(foreign)

## Loading required package: foreign

kirk <-
within(read.dta("https://stats.idre.ucla.edu/stat/stata/exa
mples/kirk/rb4.dta"),
{
s <- as.factor(s)
a <- as.factor(a)
})

model <- lm(y ~ a + s, data = kirk)
analysis <- Anova(model, idata = kirk, idesign = ~s)
print(analysis)

## Anova Table (Type II tests)
##
## Response: y
## Sum Sq Df F value Pr(>F)
```

a 49.0 3 11.6 0.00011 ***
s 31.5 7 3.2 0.01802 *
Residuals 29.5 21

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Repeated measures logistic regression

require(lme4)

Loading required package: lme4

Loading required package: Matrix

Loading required package: lattice

Attaching package: 'lme4'

The following object(s) are masked from 'package:stats':
##
AIC, BIC

exercise <-

```
within(read.dta("https://stats.idre.ucla.edu/stat/stata/whatstat/exercise.dta"),
{
id <- as.factor(id)
diet <- as.factor(diet)
})
glmer(highpulse ~ diet + (1 | id), data = exercise, family = binomial)

## Generalized linear mixed model fit by the Laplace approximation
## Formula: highpulse ~ diet + (1 | id)
## Data: exercise
## AIC BIC logLik deviance
## 105 113 -49.7 99.5
## Random effects:
## Groups Name Variance Std.Dev.
## id (Intercept) 3.32 1.82
## Number of obs: 90, groups: id, 30
##
## Fixed effects:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.004 0.663 -3.02 0.0025 **
## diet2 1.145 0.898 1.27 0.2022
```

```
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr)
## diet2 -0.738
```

Factorial ANOVA

```
anova(lm(write ~ female * ses, data = hsb2))
```

```
## Analysis of Variance Table
##
## Response: write
## Df Sum Sq Mean Sq F value Pr(>F)
## female 1 1176 1176 14.7 0.00017 ***
## ses 1 1042 1042 13.1 0.00039 ***
## female:ses 1 0 0 0.0 0.98276
## Residuals 196 15660 80
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Friedman test

```
friedman.test(cbind(read, write, math))
```

```
##
## Friedman rank sum test
##
## data: cbind(read, write, math)
## Friedman chi-squared = 0.645, df = 2, p-value = 0.7244
```

Factorial logistic regression

```
summary(glm(female ~ prog * schtyp, data = hsb2, family = binomial))
```

```
##
## Call:
## glm(formula = female ~ prog * schtyp, family = binomial, data = hsb2)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -1.89 -1.25 1.06 1.11 1.20
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.0513 0.3204 -0.16 0.87
```

prog2 0.3246 0.3911 0.83 0.41
prog3 0.2183 0.4319 0.51 0.61
schtyp2 1.6607 1.1413 1.46 0.15
prog2:schtyp2 -1.9340 1.2327 -1.57 0.12
prog3:schtyp2 -1.8278 1.8402 -0.99 0.32
##
(Dispersion parameter for binomial family taken to be 1)
##
Null deviance: 275.64 on 199 degrees of freedom
Residual deviance: 272.49 on 194 degrees of freedom
AIC: 284.5
##
Number of Fisher Scoring iterations: 3

Correlation

cor(read, write)

0.597

cor.test(read, write)

```
##
## Pearson's product-moment correlation
##
## data: read and write
## t = 10.5, df = 198, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.499 0.679
## sample estimates:
## cor
## 0.597
```

Simple linear regression

```
lm(write ~ read)
```

```
##
## Call:
## lm(formula = write ~ read)
##
## Coefficients:
## (Intercept) read
## 23.959 0.552
```

Non-parametric correlation

```
cor.test(write, read, method = "spearman")

## Warning: Cannot compute exact p-values with ties

##
## Spearman's rank correlation rho
##
## data: write and read
## S = 510993, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
## 0.617
```

Simple logistic regression

```
glm(female ~ read, family = binomial)

##
## Call: glm(formula = female ~ read, family = binomial)
##
## Coefficients:
## (Intercept) read
```

0.7261 -0.0104

##

Degrees of Freedom: 199 Total (i.e. Null); 198 Residual

Null Deviance: 276

Residual Deviance: 275 AIC: 279

Multiple regression

lm(write ~ female + read + math + science + socst)

##

Call:

lm(formula = write ~ female + read + math + science + socst)

##

Coefficients:

(Intercept) female read math science

6.139 5.493 0.125 0.238 0.242

socst

0.229

Analysis of covariance

summary(aov(write ~ prog + read))

Df Sum Sq Mean Sq F value Pr(>F)
prog 2 3176 1588 28.6 1.2e-11 ***
read 1 3842 3842 69.3 1.4e-14 ***
Residuals 196 10861 55

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Multiple logistic regression

glm(female ~ read + write, family = binomial)

##
Call: glm(formula = female ~ read + write, family = binomial)
##
Coefficients:
(Intercept) read write
-1.706 -0.071 0.106
##
Degrees of Freedom: 199 Total (i.e. Null); 197 Residual
Null Deviance: 276
Residual Deviance: 248 AIC: 254

Ordered logistic regression

Ordered logistic regression is used when the dependent variable is ordered, but not continuous. For example, using the hsb2 data file we will create an ordered variable called write3. This variable will have the values 1, 2 and 3, indicating a low, medium or high writing score. We do not generally recommend categorizing a continuous variable in this way; we are simply creating a variable to use for this example. We will use gender (female), reading score (read) and social studies score (socst) as predictor variables in this model.

```
require(MASS)

## Creat order variable write3 as a factor with levels 1, 2, and 3hsb2$write3<-cut(hsb2$write,c(0,48,57,70),right=TRUE,labels=c(1,2,3))
table(hsb2$write3)
```

##

1 2 3

61 61 78

fit ordered logit model and store results 'm'm<- polr(write3~female+read+socst,data= hsb2,Hess=TRUE)## view a summary of the modelsummary(m)

Call:

polr(formula = write3 ~ female + read + socst, data = hsb2, Hess = TRUE)

##

Coefficients:

Value Std. Error t value

female 1.2854 0.3244 3.96

read 0.1177 0.0214 5.51

socst 0.0802 0.0194 4.12

##

Intercepts:

Value Std. Error t value

1|2 9.704 1.197 8.108

2|3 11.800 1.304 9.049

##

Residual Deviance: 312.55

AIC: 322.55

Discriminant analysis

Discriminant analysis is used when you have one or more normally distributed interval independent variables and a categorical dependent variable. It is a multivariate technique that considers the latent dimensions in the independent variables for predicting group membership in the categorical dependent variable. For example, using the hsb2 data say we wish to use read, write and math scores to predict the type of program a student belongs to (prog).

```
require(MASS)

fit<-lda(factor(prog)~read+write+math,data= hsb2)fit# show results
```

```
## Call:
## lda(factor(prog) ~ read + write + math, data = hsb2)
##
## Prior probabilities of groups:
## 1 2 3
## 0.225 0.525 0.250
##
## Group means:
## read write math
## 1 49.8 51.3 50.0
## 2 56.2 56.3 56.7
## 3 46.2 46.8 46.4
##
## Coefficients of linear discriminants:
## LD1 LD2
## read 0.0292 0.0439
## write 0.0383 -0.1370
## math 0.0703 0.0793
##
## Proportion of trace:
## LD1 LD2
## 0.9874 0.0126
```

One-way MANOVA

```
summary(manova(cbind(read, write, math) ~ prog))

## Df Pillai approx F num Df den Df Pr(>F)
## prog 2 0.267 10.1 6 392 2.3e-10 ***
## Residuals 197
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Multivariate multiple regression

```
M1 <- lm(cbind(write, read) ~ female + math + science +
socst, data = hsb2)

require(car)
summary(Anova(M1))

##
## Type II MANOVA Tests:
##
## Sum of squares and products for error:
## write read
## write 7259 1091
## read 1091 8700
```

```
##
## ----------------------------------------
##
## Term: female
##
## Sum of squares and products for the hypothesis:
## write read
## write 1414 -133.5
## read -133 12.6
##
## Multivariate Tests: female
## Df test stat approx F num Df den Df Pr(>F)
## Pillai 1 0.170 19.9 2 194 1.4e-08 ***
## Wilks 1 0.830 19.9 2 194 1.4e-08 ***
## Hotelling-Lawley 1 0.205 19.9 2 194 1.4e-08 ***
## Roy 1 0.205 19.9 2 194 1.4e-08 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ----------------------------------------
##
## Term: math
##
## Sum of squares and products for the hypothesis:
```

```
## write read
## write 715 856
## read 856 1026
##
## Multivariate Tests: math
## Df test stat approx F num Df den Df Pr(>F)
## Pillai 1 0.16 18.5 2 194 4.6e-08 ***
## Wilks 1 0.84 18.5 2 194 4.6e-08 ***
## Hotelling-Lawley 1 0.19 18.5 2 194 4.6e-08 ***
## Roy 1 0.19 18.5 2 194 4.6e-08 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------
##
## Term: science
##
## Sum of squares and products for the hypothesis:
## write read
## write 858 901
## read 901 947
##
## Multivariate Tests: science
## Df test stat approx F num Df den Df Pr(>F)
```

```
## Pillai 1 0.166 19.4 2 194 2.1e-08 ***
## Wilks 1 0.834 19.4 2 194 2.1e-08 ***
## Hotelling-Lawley 1 0.200 19.4 2 194 2.1e-08 ***
## Roy 1 0.200 19.4 2 194 2.1e-08 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## ------------------------------------------
##
## Term: socst
##
## Sum of squares and products for the hypothesis:
## write read
## write 1106 1277
## read 1277 1476
##
## Multivariate Tests: socst
## Df test stat approx F num Df den Df Pr(>F)
## Pillai 1 0.221 27.5 2 194 3.1e-11 ***
## Wilks 1 0.779 27.5 2 194 3.1e-11 ***
## Hotelling-Lawley 1 0.283 27.5 2 194 3.1e-11 ***
## Roy 1 0.283 27.5 2 194 3.1e-11 ***
## ---
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Canonical correlation

require(CCA)

Loading required package: CCA

Loading required package: fda

Loading required package: zoo

Attaching package: 'zoo'

The following object(s) are masked from 'package:base':

##

as.Date, as.Date.numeric

Attaching package: 'fda'

The following object(s) are masked from 'package:graphics':

##

```
## matplot

## Loading required package: fields

## Loading required package: spam

## Spam version 0.29-3 (2013-04-23) is loaded. Type 'help( Spam)' or 'demo(
## spam)' for a short introduction and overview of this package. Help for
## individual functions is also obtained by adding the suffix '.spam' to the
## function name, e.g. 'help( chol.spam)'.

## Attaching package: 'spam'

## The following object(s) are masked from 'package:base':
##
## backsolve, forwardsolve

## Loading required package: maps
```

```
cc(cbind(read, write), cbind(math, science))

## $cor
## 0.7728 0.0235
##
## $names
## $names$Xnames
## "read" "write"
##
## $names$Ynames
## "math" "science"
##
## $names$ind.names
## NULL
##
##
## $xcoef
##
## read -0.0633 -0.104
## write -0.0492 0.122
##
## $ycoef
##
## math -0.0670 0.120
```

```
## science -0.0482 -0.121
##
## $scores
## $scores$xscores
##
## -0.2636 -0.58956
## -1.3042 -0.87790
## 1.4945 -1.55654
## -0.2492 -2.18757
## 0.3690 0.44835
## 0.5588 0.75972
## -0.1655 0.99033
## 1.4869 1.06618
## -0.8894 -0.60276
## -0.4113 -0.22384
## -0.1579 -1.63238
## -0.9038 0.99525
## -1.6698 -1.27495
## -0.6155 1.06280
## 0.2493 1.26547
## 0.8331 0.60158
## 0.3690 0.44835
## -0.5098 0.01998
## -1.5997 -0.14645
```

0.5032 -1.96679
-0.4954 -1.57803
-1.1849 0.12869
0.7702 -1.32593
-0.4534 -0.90093
1.3956 0.51099
1.9302 -0.03100
-1.4099 0.16492
0.1228 1.05789
1.2483 -0.94700
0.4467 -1.04587
1.7196 -1.59277
-1.4656 -2.56159
-1.5084 0.40874
1.2271 -0.37369
-0.2920 0.78275
-1.0936 0.68388
-1.0580 -1.48744
-1.2622 -0.20080
1.4033 -2.11173
1.9093 -1.28140
0.6153 -0.16119
-0.4818 0.47138
-0.0458 0.17321

```
## 1.2271 -0.37369
## -1.4099 0.16492
## -0.4818 0.47138
## 0.7702 -1.32593
## -1.1144 -0.56653
## 0.0243 1.30171
## -0.3621 -0.34574
## -0.4538 0.92278
## 1.8108 -1.03759
## 0.9528 -0.21555
## -0.9879 -0.35895
## -1.7683 -1.03113
## 1.5154 -0.30614
## 1.7404 -0.34237
## 1.3256 -0.61751
## -0.8894 -0.60276
## 0.4535 -0.02117
## 0.4747 -0.59448
## -0.1235 1.66743
## 0.9528 -0.21555
## 2.1272 -0.51863
## 0.6717 -1.08211
## 1.1005 -0.58127
## -0.5519 -0.65712
```

1.0093 -1.13646
-0.1999 -2.30948
1.1150 -2.17928
0.9528 -0.21555
-0.5519 -0.65712
-2.0145 -0.42159
-1.3042 -0.87790
0.2077 -1.23534
0.9600 -1.01455
-0.5803 0.71520
-1.3042 -0.87790
2.1692 0.15847
0.9108 -0.89265
-0.9879 -0.35895
1.2126 1.22432
-1.3042 -0.87790
-1.2834 0.37250
0.4047 -1.72297
-0.6296 0.83710
0.5945 -1.41160
1.3392 1.43190
1.2135 -2.42310
0.0107 -0.74770
-0.4398 1.14848

-0.4954 -1.57803

-1.4520 -0.51218

1.2691 0.30341

0.9528 -0.21555

-0.3133 1.35606

-1.7895 -0.45782

-1.2834 0.37250

1.5854 0.82236

-1.1849 0.12869

-1.3535 -0.75599

0.0243 1.30171

0.6645 -0.28310

-0.6432 -1.21231

-0.2920 0.78275

-0.2356 -0.13816

-0.9459 0.31815

1.9654 -0.37861

0.2705 0.69216

-1.7895 -0.45782

-0.2636 -0.58956

0.6573 0.51590

-1.1144 -0.56653

-1.5997 -0.14645

-1.7190 -1.15304

1.4589 0.61478

0.5236 1.10733

-2.0145 -0.42159

-0.1935 0.53893

0.9948 0.46155

-0.5519 -0.65712

0.1792 0.13697

0.1792 0.13697

0.6645 -0.28310

-0.1235 1.66743

-0.3833 0.22756

0.7210 -1.20402

0.8259 1.40058

0.3270 -0.22875

2.0287 -0.27481

-0.6083 0.26380

-0.9038 0.99525

-1.4520 -0.51218

0.5868 1.21112

-0.8614 -0.15137

-2.0073 -1.22059

0.0243 1.30171

0.4323 0.55214

1.4169 -0.06232

0.2005 -0.43633
1.8665 1.68892
0.5868 1.21112
0.8615 -0.77074
0.1228 1.05789
-0.2920 0.78275
0.3690 0.44835
-0.3133 1.35606
0.5588 0.75972
0.9108 -0.89265
0.3478 1.02165
1.1078 -1.38028
-0.8682 -1.17607
0.3758 1.47305
0.2705 0.69216
-0.7561 0.62952
-1.3042 -0.87790
-0.0950 0.29512
0.4391 1.57684
1.3256 -0.61751
-1.5717 0.30495
-0.1235 1.66743
-0.1651 -0.83338
-0.0742 1.54552

-0.7561 0.62952

-0.2920 0.78275

0.9528 -0.21555

-0.1655 0.99033

0.7766 1.52249

-0.7561 0.62952

-0.6576 0.38571

0.4391 1.57684

0.6645 -0.28310

1.4733 -0.98323

-0.7981 -0.04757

0.7066 0.39399

-1.0371 -0.23704

-1.5084 0.40874

0.7766 1.52249

0.1792 0.13697

-0.5875 1.51420

-0.9459 0.31815

0.7066 0.39399

-0.6860 1.75802

-0.7773 1.20283

-0.5595 1.96560

-1.4099 0.16492

-0.0458 0.17321

```
## 0.7630 -0.52692
## -1.1356 0.00678
## 0.4747 -0.59448
## 0.5868 1.21112
## 0.8187 2.19959
## 0.1792 0.13697
## 0.4038 1.92445
## -0.2712 2.03316
## -0.4818 0.47138
## 0.9808 0.23585
## 0.2782 -1.93055
## -0.6296 0.83710
## -1.2834 0.37250
##
## $scores$yscores
##
## 1.01398 -0.8128
## -0.56166 -1.3052
## -0.38744 -0.5807
## 0.32264 -0.8172
## -0.34719 0.3842
## -0.42770 -1.5455
## 0.65755 -1.4179
## 1.13197 0.6349
```

```
## -0.38744 -0.5807
## 0.13245 0.1461
## 0.05471 -0.3367
## -0.42770 -1.5455
## -1.67087 1.0991
## -0.44367 0.1424
## 1.18299 2.2027
## 0.73529 -0.9351
## 0.19943 0.0260
## -0.78934 0.1402
## -0.77858 0.7431
## -0.34719 0.3842
## 0.49930 -3.8305
## -2.46945 0.2499
## 0.36012 -1.2993
## -0.73311 -0.5829
## 1.13197 0.6349
## 1.06499 0.7550
## -1.33596 0.4984
## -0.58839 -0.2202
## 0.86404 1.1155
## 0.09219 -0.8187
## -0.05774 1.1095
## -1.34671 -0.1045
```

```
## -1.37621 -0.4665
## -1.26376 -1.9126
## -0.26423 -1.4239
## -0.80009 -0.4627
## -2.18000 0.9752
## -1.71112 0.1343
## 1.34368 0.8774
## 1.46689 0.0342
## 1.25518 -0.2083
## -0.92330 0.3805
## -0.44367 0.1424
## 0.73529 -0.9351
## -0.22675 -1.9059
## -1.52093 -0.8291
## 1.05146 -1.2948
## -1.70037 0.7372
## -0.74908 1.1051
## -0.19171 1.3498
## -0.80808 0.3812
## 1.21493 -1.1732
## -0.47040 1.2274
## 0.09219 -0.8187
## -2.19076 0.3723
## 1.82609 2.0863
```

```
## 1.62237 0.9998
## 0.71378 -2.1410
## -0.45442 -0.4605
## 0.42189 0.8715
## 0.21540 -1.6619
## -1.38697 -1.0694
## 1.77507 0.5185
## 1.54463 0.5170
## 0.94178 1.5983
## 0.93624 -1.2956
## -0.64462 0.5029
## 0.85329 0.5125
## -1.06004 -0.8261
## 0.62284 0.5110
## 1.06499 0.7550
## -0.65537 -0.1001
## -1.76735 0.8573
## -1.42722 -2.0343
## 0.14842 -1.5418
## 0.97650 -0.3307
## 0.04672 0.5073
## -0.76261 -0.9448
## 1.06499 0.7550
## 0.38441 1.3535
```

```
## -0.44367 0.1424
## -0.53216 -0.9433
## -1.91207 0.4947
## -0.22675 -1.9059
## 1.22568 -0.5702
## -1.29847 0.0163
## 0.05471 -0.3367
## 1.38915 -0.4486
## 1.70809 0.6386
## -1.00104 -0.1023
## -0.66059 2.1908
## -0.43845 -2.1485
## -1.65490 -0.5888
## 0.42189 0.8715
## 0.67907 -0.2120
## -1.09752 -0.3441
## -1.97905 0.6148
## -2.05679 0.1320
## 1.46689 0.0342
## -1.53690 0.8588
## -1.58792 -0.7090
## -0.90733 -1.3075
## 1.15349 1.8408
## -0.00152 0.3864
```

```
## -0.46518 -1.0635
## -0.41938 2.7952
## -0.87229 1.9483
## 0.88800 -1.4164
## 0.53435 -0.5747
## -1.39218 1.2215
## 0.40559 -2.6253
## 1.39991 0.1543
## -0.00950 1.2304
## -0.70361 -0.2210
## -0.44367 0.1424
## 1.52311 -0.6889
## -0.30970 -0.0979
## -0.92330 0.3805
## -1.05727 0.6208
## 1.46689 0.0342
## 0.26641 -0.0941
## 0.53435 -0.5747
## 0.59611 1.5960
## 0.22893 0.3879
## 1.37318 1.2393
## -0.52141 -0.3404
## 0.89077 0.0305
## -0.00152 0.3864
```

0.00924 0.9894
1.88231 1.3632
-0.00152 0.3864
-1.40017 2.0655
-0.13548 0.6267
0.61208 -0.0919
0.62284 0.5110
-1.22350 -0.9478
-0.38744 -0.5807
0.22094 1.2319
1.79104 -1.1695
0.62284 0.5110
1.02474 -0.2098
0.26641 -0.0941
0.66310 1.4759
0.68982 0.3909
-0.41417 0.5043
0.83177 -0.6934
0.62806 -1.7798
1.02474 -0.2098
1.01675 0.6342
1.44016 1.1192
1.12122 0.0320
-0.85632 0.2603

```
## 0.93624 -1.2956
## 0.18868 -0.5769
## -0.52141 -0.3404
## -0.71160 0.6230
## 0.07345 -0.5777
## 0.34415 0.3887
## 1.18820 -0.0882
## -1.40294 0.6185
## -0.07926 -0.0964
## 0.21817 -0.2150
## -0.42770 -1.5455
## -1.73785 1.2193
## 0.15918 -0.9389
## 1.41865 -0.0867
## -0.46518 -1.0635
## 1.14795 -1.0530
## -0.84556 0.8633
## 0.05471 -0.3367
## 0.60133 -0.6948
## 1.14795 -1.0530
## 1.71885 1.2416
## -1.06802 0.0178
## 1.39192 0.9983
## -0.93129 1.2245
```

```
## -0.84556 0.8633
## -0.14624 0.0238
## 0.21540 -1.6619
## -0.69286 0.3820
## 0.33340 -0.2143
## 0.93103 0.9953
## -0.82959 -0.8247
## -0.06850 0.5066
## -1.57716 -0.1060
## -1.05727 0.6208
## -0.21322 0.1439
## 1.18820 -0.0882
## -0.37668 0.0223
## -0.07926 -0.0964
## 1.25518 -0.2083
## 0.80228 -1.0553
## -0.17574 -0.3381
## 1.57413 0.8789
## -0.40341 1.1073
## 0.51837 1.1132
## 1.74558 0.1566
## -0.44367 0.1424
## -0.65537 -0.1001
## -0.88305 1.3453
```

```
##
## $scores$corr.X.xscores
##
## read -0.927 -0.375
## write -0.854 0.520
##
## $scores$corr.Y.xscores
##
## math -0.718 0.0087
## science -0.675 -0.0114
##
## $scores$corr.X.yscores
##
## read -0.717 -0.00879
## write -0.660 0.01222
##
## $scores$corr.Y.yscores
##
## math -0.929 0.371
## science -0.873 -0.487
```

Factor analysis

```
require(psych)
```

```
fa(r = cor(model.matrix(~read + write + math + science +
socst - 1, data = hsb2)), rotate = "none", fm = "pa", 2)

## Factor Analysis using method = pa
## Call: fa(r = cor(model.matrix(~read + write + math +
science + socst -
## 1, data = hsb2)), nfactors = 2, rotate = "none", fm =
"pa")
## Standardized loadings (pattern matrix) based upon
correlation matrix
## PA1 PA2 h2 u2 com
## read 0.81 0.06 0.66 0.34 1.0
## write 0.76 0.00 0.58 0.42 1.0
## math 0.80 0.17 0.67 0.33 1.1
## science 0.75 0.26 0.62 0.38 1.2
## socst 0.79 -0.48 0.85 0.15 1.6
##
## PA1 PA2
## SS loadings 3.06 0.33
## Proportion Var 0.61 0.07
## Cumulative Var 0.61 0.68
## Proportion Explained 0.90 0.10
## Cumulative Proportion 0.90 1.00
##
```

Mean item complexity = 1.2
Test of the hypothesis that 2 factors are sufficient.
##
The degrees of freedom for the null model are 10 and the objective function was 2.51
The degrees of freedom for the model are 1 and the objective function was 0.01
##
The root mean square of the residuals (RMSR) is 0.01
The df corrected root mean square of the residuals is 0.03
##
Fit based upon off diagonal values = 1
Measures of factor score adequacy
PA1 PA2
Correlation of scores with factors 0.95 0.79
Multiple R square of scores with factors 0.91 0.62
Minimum correlation of possible factor scores 0.82 0.23

Principal components analysis

princomp(formula = ~read + write + math + science + socst, data = hsb2)

```
## Call:
## princomp(formula = ~read + write + math + science + socst, data = hsb2)
##
## Standard deviations:
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## 18.252929 7.677044 6.213371 5.774331 5.429881
##
## 5 variables and 200 observations.
```