Booklet of Code and Output
for
STAD29/STA 1007 Final Exam

List of Figures in this document by page:

List of Figures

```
gorilla

##    seen   W   C CW
## 1     0 126  86 64
## 2     0 118  76 54
## 3     0  61  66 44
## 4     0  69  48 32
## 5     0  57  59 42
## 6     0  78  64 53
## 7     0 114  61 41
## 8     0  81  85 47
## 9     0  73  57 33
## 10    0  93  50 45
## 11    0 116  92 49
## 12    0 156  70 45
## 13    0  90  66 48
## 14    0 120  73 49
## 15    0  99  68 44
## 16    0 113 110 47
## 17    0 103  78 52
## 18    0 123  61 28
## 19    0  86  65 42
## 20    0  99  77 51
## 21    0 102  77 54
## 22    0 120  74 53
## 23    0 128 100 56
## 24    0 100  89 56
## 25    0  95  61 37
## 26    0  80  55 36
## 27    0  98  92 51
## 28    0 111  90 52
## 29    0 101  85 45
## 30    0 102  78 51
## 31    1 100  66 48
## 32    1 112  78 55
## 33    1  82  84 37
## 34    1  72  63 46
## 35    1  72  65 47
## 36    1  89  71 49
## 37    1 108  46 29
## 38    1  88  70 49
## 39    1 116  83 67
## 40    1 100  69 39
## 41    1  99  70 43
## 42    1  93  63 36
## 43    1 100  93 62
## 44    1 110  76 56
## 45    1 100  83 36
## 46    1 106  71 49
## 47    1 115 112 66
## 48    1 120  87 54
## 49    1  97  82 41
```

Figure 1: Gorilla and Stroop word test data

```r
gorilla.0=glm(seen~1,data=gorilla,family="binomial")
gorilla.1=glm(seen~W+C+CW,data=gorilla,family="binomial")
summary(gorilla.1)

##
## Call:
## glm(formula = seen ~ W + C + CW, family = "binomial", data = gorilla)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0928  -1.0129  -0.9302   1.3330   1.6281
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.755980   1.968679  -0.384    0.701
## W           -0.009506   0.018147  -0.524    0.600
## C            0.007602   0.027876   0.273    0.785
## CW           0.014361   0.044204   0.325    0.745
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 65.438  on 48  degrees of freedom
## Residual deviance: 64.946  on 45  degrees of freedom
## AIC: 72.946
##
## Number of Fisher Scoring iterations: 4

anova(gorilla.0,gorilla.1)

## Analysis of Deviance Table
##
## Model 1: seen ~ 1
## Model 2: seen ~ W + C + CW
##   Resid. Df Resid. Dev Df Deviance
## 1        48     65.438
## 2        45     64.946  3  0.49144
```

Figure 2: Logistic regression for gorilla data

```r
boxplot(CW~seen,data=gorilla)
```

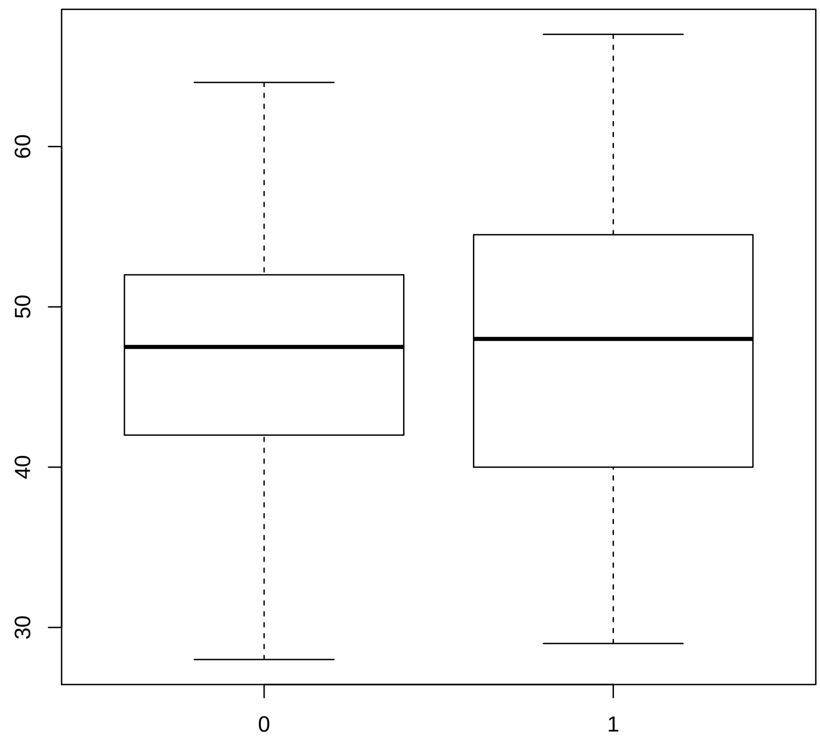

Figure 3: Boxplot of Stroop colour-and-word scores by whether or not gorilla seen

```r
autos=read.table("autos.txt",header=T)
head(autos,n=20)
```

```
##                make price mpg rep78 headroom trunk weight length turn
## 1       AMC Concord  4099  22     3      2.5    11   2930    186   40
## 2         AMC Pacer  4749  17     3      3.0    11   3350    173   40
## 3        AMC Spirit  3799  22    NA      3.0    12   2640    168   35
## 4     Buick Century  4816  20     3      4.5    16   3250    196   40
## 5     Buick Electra  7827  15     4      4.0    20   4080    222   43
## 6     Buick LeSabre  5788  18     3      4.0    21   3670    218   43
## 7        Buick Opel  4453  26    NA      3.0    10   2230    170   34
## 8       Buick Regal  5189  20     3      2.0    16   3280    200   42
## 9     Buick Riviera 10372  16     3      3.5    17   3880    207   43
## 10    Buick Skylark  4082  19     3      3.5    13   3400    200   42
## 11     Cad. Deville 11385  14     3      4.0    20   4330    221   44
## 12    Cad. Eldorado 14500  14     2      3.5    16   3900    204   43
## 13     Cad. Seville 15906  21     3      3.0    13   4290    204   45
## 14   Chev. Chevette  3299  29     3      2.5     9   2110    163   34
## 15     Chev. Impala  5705  16     4      4.0    20   3690    212   43
## 16     Chev. Malibu  4504  22     3      3.5    17   3180    193   31
## 17 Chev. Monte Carlo  5104  22     2      2.0    16   3220    200   41
## 18      Chev. Monza  3667  24     2      2.0     7   2750    179   40
## 19       Chev. Nova  3955  19     3      3.5    13   3430    197   43
## 20      Dodge Colt  3984  30     5      2.0     8   2120    163   35
##    displacement gear_ratio
## 1           121       3.58
## 2           258       2.53
## 3           121       3.08
## 4           196       2.93
## 5           350       2.41
## 6           231       2.73
## 7           304       2.87
## 8           196       2.93
## 9           231       2.93
## 10          231       3.08
## 11          425       2.28
## 12          350       2.19
## 13          350       2.24
## 14          231       2.93
## 15          250       2.56
## 16          200       2.73
## 17          200       2.73
## 18          151       2.73
## 19          250       2.56
## 20           98       3.54
```

Figure 4: 1978 cars data (some)

```
library(MASS)
repf=ordered(autos$rep78)
autos.1=polr(repf~price+mpg+headroom+trunk+weight+length+turn+
    displacement+gear_ratio,data=autos)
autos.2=polr(repf ~ price + mpg + length + turn + displacement,
    data = autos)
anova(autos.2,autos.1)

## Likelihood ratio tests of ordinal regression models
##
## Response: repf
##                                                                                  Model
## 1                                          price + mpg + length + turn + displacement
## 2 price + mpg + headroom + trunk + weight + length + turn + displacement + gear_ratio
##   Resid. df Resid. Dev   Test    Df LR stat.  Pr(Chi)
## 1        60   158.0758
## 2        56   156.5030 1 vs 2     4 1.572858 0.813662
```

Figure 5: Cars fitted models

```
prices=c(4000,6000)
mpgs=20
lengths=200
turns=c(35,45)
displacements=200
new=expand.grid(price=prices,mpg=mpgs,length=lengths,turn=turns,
  displacement=displacements)
pp=predict(autos.2,new,type="p")
cbind(new,pp)

##   price mpg length turn displacement           1          2         3
## 1  4000  20    200   35          200 0.003569743 0.01942962 0.2391378
## 2  6000  20    200   35          200 0.002311087 0.01268201 0.1718085
## 3  4000  20    200   45          200 0.055555939 0.22322005 0.5748827
## 4  6000  20    200   45          200 0.036641509 0.16331294 0.5904809
##           4          5
## 1 0.4359874 0.30187542
## 2 0.4124494 0.40074898
## 3 0.1206823 0.02565910
## 4 0.1704300 0.03913469
```

Figure 6: Predictions of repair record for cars data

```
gbcs=read.table("gbcs.txt",header=T)
str(gbcs)

## 'data.frame': 686 obs. of  16 variables:
##  $ id        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ diagdateb : Factor w/ 525 levels "1984-04-25","1984-05-
29",..: 21 93 38 9 10 14 109 29 43 106 ...
##  $ recdate   : Factor w/ 494 levels "1984-11-24","1985-01-
18",..: 151 211 149 1 240 268 27 365 454 471 ...
##  $ deathdate : Factor w/ 471 levels "1984-11-24","1985-03-
08",..: 284 275 117 1 174 200 16 390 423 440 ...
##  $ age       : int  38 52 47 40 64 49 53 61 43 74 ...
##  $ menopause : int  1 1 1 1 2 2 2 2 1 2 ...
##  $ hormone   : int  1 1 1 1 2 2 1 2 1 2 ...
##  $ size      : int  18 20 30 24 19 56 52 22 30 20 ...
##  $ grade     : int  3 1 2 1 2 1 2 2 2 2 ...
##  $ nodes     : int  5 1 1 3 1 3 9 2 1 1 ...
##  $ prog_recp : int  141 78 422 25 19 356 6 6 22 462 ...
##  $ estrg_recp: int  105 14 89 11 9 64 29 173 0 240 ...
##  $ rectime   : int  1337 1420 1279 148 1863 1933 358 2372 2563 2372 ...
##  $ censrec   : int  1 1 1 0 0 0 1 1 0 0 ...
##  $ survtime  : int  2282 2006 1456 148 1863 1933 416 2556 2563 2372 ...
##  $ censdead  : int  0 0 1 0 0 0 1 0 0 0 ...
```

Figure 7: Breast cancer data

```r
attach(gbcs)
library(survival)
y=Surv(rectime,censrec==1)
y.1=coxph(y~hormone+size+nodes)
summary(y.1)

## Call:
## coxph(formula = y ~ hormone + size + nodes)
##
##   n= 686, number of events= 299
##
##              coef exp(coef)  se(coef)      z Pr(>|z|)
## hormone -0.364145  0.694790  0.125258 -2.907  0.00365 **
## size     0.007426  1.007454  0.003854  1.927  0.05399 .
## nodes    0.052367  1.053763  0.007319  7.155 8.39e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##         exp(coef) exp(-coef) lower .95 upper .95
## hormone    0.6948     1.4393    0.5435    0.8881
## size       1.0075     0.9926    0.9999    1.0151
## nodes      1.0538     0.9490    1.0388    1.0690
##
## Concordance= 0.657  (se = 0.016 )
## Likelihood ratio test= 62.02  on 3 df,   p=2e-13
## Wald test            = 87.73  on 3 df,   p=<2e-16
## Score (logrank) test = 90.83  on 3 df,   p=<2e-16
```

Figure 8: Cox modelling of recurrence time

```r
hormones=c(1,2)
sizes=c(20,35)
nodeses=c(1,7)
new=expand.grid(hormone=hormones,size=sizes,nodes=nodeses)
pp=survfit(y.1,new)
```

Figure 9: “Survfit”

```
combo=apply(new,1,paste,collapse="-")
combo

## [1] "1-20-1" "2-20-1" "1-35-1" "2-35-1" "1-20-7" "2-20-7" "1-
35-7" "2-35-7"

v=data.frame(time=pp$time,surv=pp$surv)
names(v)=c("time",combo)
library(tidyr)

## Warning: package 'tidyr' was built under R version 3.5.3

p=gather(v,combo,surv,-time)
```

Figure 10: Code for making a ggplot of survival curves

```r
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.3

ggplot(p,aes(x=time,y=surv,colour=combo))+geom_point()+geom_line()
```

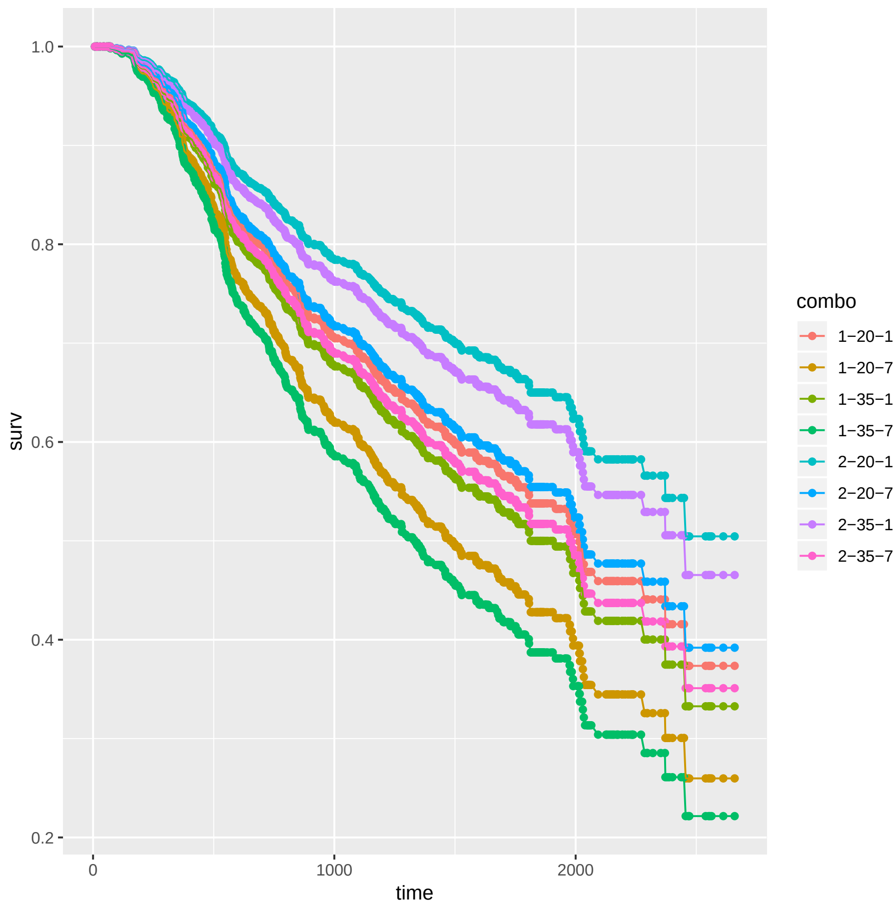

Figure 11: Plot of survival curves

```
french=read.table("learning-french.txt",header=T)
french

##    student book test  gpa
## 1        1    a   34 3.24
## 2        2    a   56 3.44
## 3        3    a   64 3.54
## 4        4    a   69 3.59
## 5        5    a   77 3.65
## 6        6    b   46 3.28
## 7        7    b   61 3.43
## 8        8    b   66 3.48
## 9        9    b   75 3.58
## 10      10    b   77 3.63
## 11      11    c   59 3.35
## 12      12    c   61 3.39
## 13      13    c   72 3.47
## 14      14    c   76 3.52
## 15      15    c   84 3.62
```

Figure 12: French learning data

```
french.1=aov(test~book,data=french)
summary(french.1)

##             Df Sum Sq Mean Sq F value Pr(>F)
## book         2  270.5   135.3   0.759  0.489
## Residuals   12 2137.2   178.1
```

Figure 13: Analysis of variance of test score by book

```r
library(ggplot2)
ggplot(french,aes(x=gpa,y=test,colour=book))+
  geom_point()
```

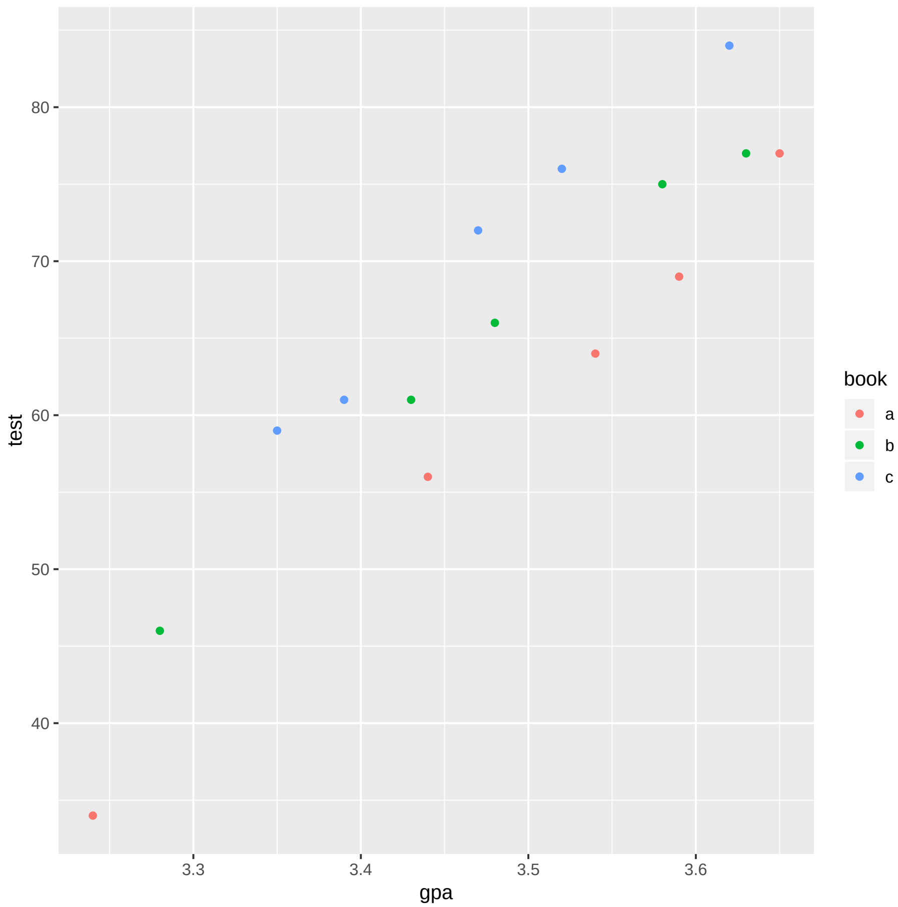

Figure 14: Scatter plot of test score against GPA labelled by textbook

```
french.2=lm(test~gpa*book,data=french)
anova(french.2)

## Analysis of Variance Table
##
## Response: test
##           Df  Sum Sq Mean Sq   F value    Pr(>F)
## gpa        1 1994.55 1994.55 1081.2054 1.095e-10 ***
## book       2  390.95  195.47  105.9621 5.559e-07 ***
## gpa:book   2    5.64    2.82    1.5275    0.2684
## Residuals  9   16.60    1.84
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

french.3=update(french.2,.~.-gpa:book)
anova(french.3)

## Analysis of Variance Table
##
## Response: test
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## gpa        1 1994.55 1994.55 986.583 4.044e-12 ***
## book       2  390.95  195.47  96.689 1.048e-07 ***
## Residuals 11   22.24    2.02
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(french.3)

##
## Call:
## lm(formula = test ~ gpa + book, data = french)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5792 -0.8223  0.2805  0.9299  1.6432
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -279.4047    10.5128 -26.578 2.48e-11 ***
## gpa           97.1949     3.0050  32.344 2.94e-12 ***
## bookb          6.1663     0.9000   6.852 2.76e-05 ***
## bookc         12.5383     0.9017  13.905 2.52e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.422 on 11 degrees of freedom
## Multiple R-squared:  0.9908,Adjusted R-squared:  0.9882
## F-statistic: 393.3 on 3 and 11 DF,  p-value: 1.816e-11
```

Figure 15: Analysis of covariance of test score by book and gpa

```
selfesteem=read.table("therapy.txt",header=T)
selfesteem

##    subject  therapy w1 w2 w3 w4
## 1        1 baseline  3  5  9  6
## 2        2 baseline  7 11 12 11
## 3        3 baseline  9 13 14 12
## 4        4 baseline  4  8 11  7
## 5        5 baseline  1  3  5  4
## 6        6      new  5  6 11  7
## 7        7      new 10 12 18 15
## 8        8      new 10 15 15 14
## 9        9      new  6  9 13  9
## 10      10      new  3  5  9  7
```

Figure 16: Self-esteem therapy data

```
attach(selfesteem)
response=cbind(w1,w2,w3,w4)
selfesteem.1=lm(response~therapy)
library(car)

## Warning: package 'car' was built under R version 3.5.1
## Loading required package: carData
## Warning: package 'carData' was built under R version 3.5.1

weeks=colnames(response)
weeks.df=data.frame(weeks)
selfesteem.2=Manova(selfesteem.1,idata=weeks.df,idesign=~weeks)
selfesteem.2

##
## Type II Repeated Measures MANOVA Tests: Pillai test statistic
##               Df test stat approx F num Df den Df    Pr(>F)
## (Intercept)    1   0.88773   63.259      1      8 4.554e-
05 ***
## therapy        1   0.10886    0.977      1      8    0.3518
## weeks          1   0.98033   99.659      3      6 1.653e-
05 ***
## therapy:weeks  1   0.25600    0.688      3      6    0.5915
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 17: Repeated measures analysis for therapy data

```r
library(tidyr)
selfesteem.long=gather(selfesteem,weeks,self.esteem,w1:w4)
```

Figure 18: Reorganizing the therapy data

```r
ggplot(selfesteem.long,
  aes(x=weeks,y=self.esteem,
  group=subject,colour=therapy)) + geom_line()
```

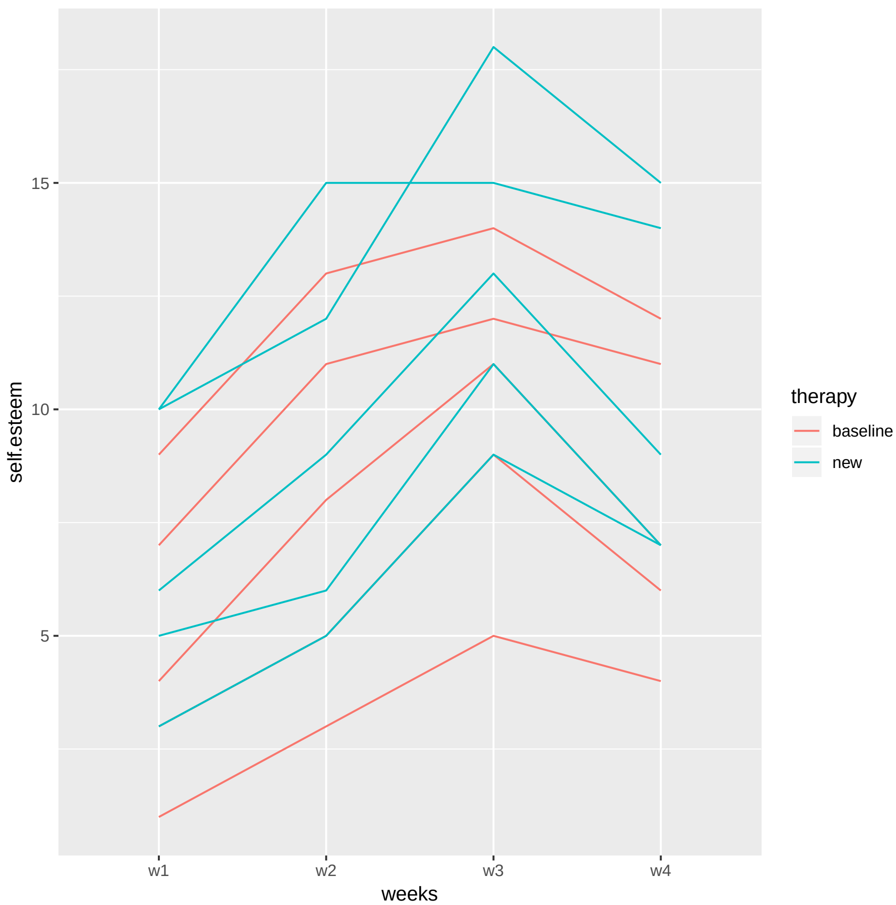

Figure 19: Spaghetti plot for therapy data

```r
mags=read.table("MAGAZINES.txt",header=T)
str(mags)

## 'data.frame': 141 obs. of  16 variables:
##  $ id       : int  101 102 103 104 105 106 107 108 109 110 ...
##  $ magazine : int  4 3 2 4 4 2 1 3 4 2 ...
##  $ i1       : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ i2       : int  0 0 1 0 0 1 0 0 0 1 ...
##  $ i3       : int  0 1 0 0 0 0 0 1 0 0 ...
##  $ i4       : int  1 0 0 1 1 0 0 0 1 0 ...
##  $ famsize  : int  2 2 1 1 2 1 2 1 1 4 ...
##  $ income   : int  11 11 8 11 10 6 11 1 6 4 ...
##  $ race     : int  1 1 1 1 1 1 1 0 1 0 ...
##  $ tv       : int  3 2 1 1 2 1 3 3 2 2 ...
##  $ newspaper: int  1 1 0 1 1 1 1 0 0 0 ...
##  $ nomale   : int  0 0 1 1 0 1 0 1 1 0 ...
##  $ nofemale : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ child18  : int  1 1 1 1 1 1 1 1 1 0 ...
##  $ headage  : int  6 5 6 5 6 6 5 5 5 3 ...
##  $ headeduc : int  7 7 4 6 6 5 7 3 6 5 ...
```

Figure 20: Summary of magazine data

```r
library(MASS)
mags.1=lda(magazine~famsize+income+race+tv+newspaper+nomale+
             nofemale+child18+headage+headeduc,data=mags)
mags.1

## Call:
## lda(magazine ~ famsize + income + race + tv + newspaper + nomale +
##     nofemale + child18 + headage + headeduc, data = mags)
##
## Prior probabilities of groups:
##         1         2         3         4
## 0.1843972 0.3475177 0.2765957 0.1914894
##
## Group means:
##    famsize   income      race       tv newspaper    nomale   nofemale
## 1 2.730769 7.384615 0.8846154 2.000000 0.4615385 0.1538462 0.00000000
## 2 2.163265 4.897959 0.7142857 1.734694 0.3877551 0.5510204 0.06122449
## 3 2.102564 5.333333 0.6923077 1.794872 0.2564103 0.3846154 0.10256410
## 4 2.074074 8.370370 0.8518519 1.703704 0.5185185 0.3703704 0.18518519
##     child18  headage headeduc
## 1 0.7307692 4.692308 5.500000
## 2 0.7755102 5.061224 4.775510
## 3 0.7435897 4.589744 5.000000
## 4 0.8148148 4.444444 6.185185
##
## Coefficients of linear discriminants:
##                   LD1         LD2         LD3
## famsize   -0.02506254  0.31004546  0.99557640
## income    -0.26488338  0.04896005  0.03142841
## race       0.02724269 -0.82876993  0.60290221
## tv         0.34729706  0.01278338 -0.22388539
## newspaper  0.12958772  0.55766570  0.97724506
## nomale    -0.06306391  2.70594323  1.26670569
## nofemale  -0.58564087  3.10963239  0.14905016
## child18   -0.75018974 -0.14545284  1.07604018
## headage    0.23540594  0.03287999  0.52175803
## headeduc  -0.27938790  0.10107380  0.01459669
##
## Proportion of trace:
##    LD1    LD2    LD3
## 0.5715 0.2739 0.1546
```

Figure 21: Discriminant analysis for magazine data

```
plot(mags.1,dimen=2,col=mags$magazine)
```

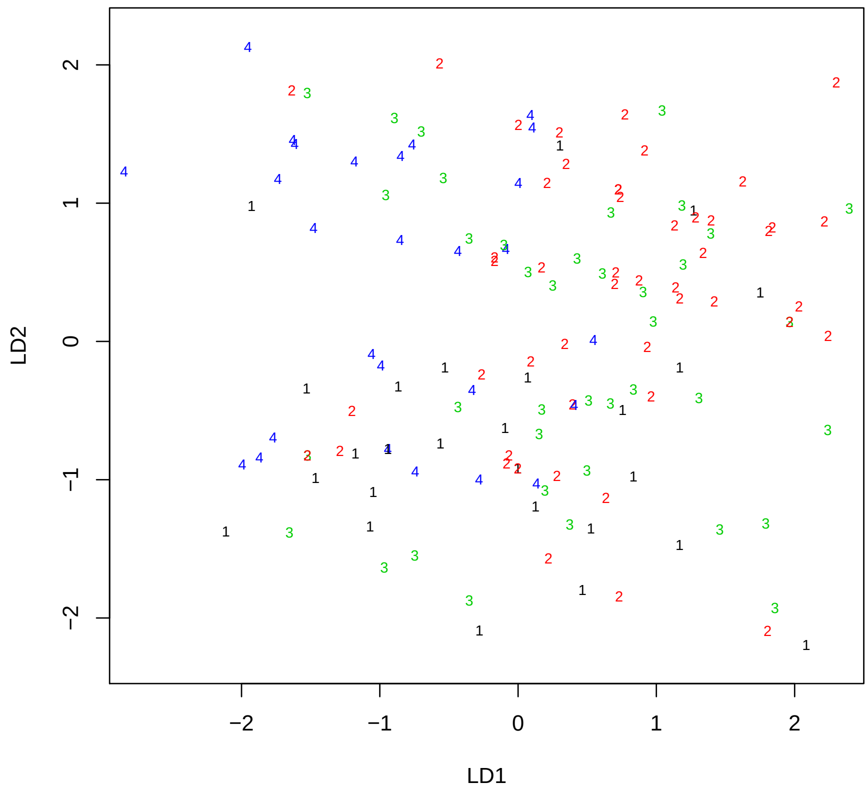

Figure 22: Plot of first two discriminant scores for magazine data

```
pp=predict(mags.1)
table(magazine=mags$magazine,predicted=pp$class)

##          predicted
## magazine  1  2  3  4
##        1 10  8  5  3
##        2  8 32  7  2
##        3  4 16 14  5
##        4  4  6  2 15
```

Figure 23: Predictions for magazine data

```
set.seed(457299)
post=round(pp$posterior,3)
row=sample(nrow(post),10)
cbind(mags[row,],guess=pp$class[row],post[row,])

##      id magazine i1 i2 i3 i4 famsize income race tv newspaper nomale
## 134 234        2  0  1  0  0       2     11    1  2         1      0
## 18  118        4  0  0  0  1       4      8    1  3         0      0
## 32  132        1  1  0  0  0       4      9    1  2         1      0
## 77  177        3  0  0  1  0       1     11    1  3         1      1
## 54  154        1  1  0  0  0       4      7    1  1         0      0
## 103 203        4  0  0  0  1       2     11    1  1         1      0
## 33  133        4  0  0  0  1       4     11    1  2         1      0
## 30  130        3  0  0  1  0       4      6    0  2         0      0
## 78  178        2  0  1  0  0       1      5    1  1         0      1
## 110 210        1  1  0  0  0       1      4    0  1         0      1
##     nofemale child18 headage headeduc guess     1     2     3     4
## 134        0       1       5        7     1 0.393 0.100 0.129 0.378
## 18         0       1       5        6     1 0.468 0.256 0.156 0.120
## 32         0       0       4        7     2 0.287 0.293 0.182 0.238
## 77         0       1       6        6     4 0.075 0.367 0.110 0.447
## 54         0       0       3        4     3 0.325 0.205 0.394 0.075
## 103        0       1       5        7     4 0.384 0.074 0.090 0.452
## 33         0       0       4        7     4 0.276 0.207 0.137 0.379
## 30         0       1       6        4     2 0.257 0.500 0.187 0.057
## 78         0       1       5        6     2 0.118 0.354 0.299 0.230
## 110        0       1       6        6     2 0.050 0.487 0.291 0.172
```

Figure 24: Sample of posterior probabilities

```
cars=read.csv("CAR_DISSIM.csv",header=T,stringsAsFactors=F)
cars

##          Car BMW Ford Infnti Jeep Lexus Chrys Merc Saab Porsche Volvo
## 1        BMW   0   NA     NA   NA    NA    NA   NA   NA      NA    NA
## 2       Ford  34    0     NA   NA    NA    NA   NA   NA      NA    NA
## 3   Infiniti   8   24      0   NA    NA    NA   NA   NA      NA    NA
## 4       Jeep  31    2     25    0    NA    NA   NA   NA      NA    NA
## 5      Lexus   7   26      1   27     0    NA   NA   NA      NA    NA
## 6   Chrysler  43   14     35   15    37     0   NA   NA      NA    NA
## 7   Mercedes   3   28      5   29     4    42    0   NA      NA    NA
## 8       Saab  10   18     20   17    13    36   19    0      NA    NA
## 9    Porsche   6   39     41   38    40    45   32   21       0    NA
## 10     Volvo  33   11     22   12    23     9   30   16      44     0
```

Figure 25: Cars data for multidimensional scaling

```
d=as.dist(cars[,-1])
library(MASS)
d.1=isoMDS(d)

## initial  value 10.466056
## iter   5 value 6.523088
## iter  10 value 4.865126
## iter  15 value 4.088399
## iter  20 value 4.003118
## final  value 3.989006
## converged

d.1$stress

## [1] 3.989006
```

Figure 26: Multidimensional scaling analysis of cars

```
plot(d.1$points,xlim=c(-30,30))
text(d.1$points,cars$Car,pos=4)
```

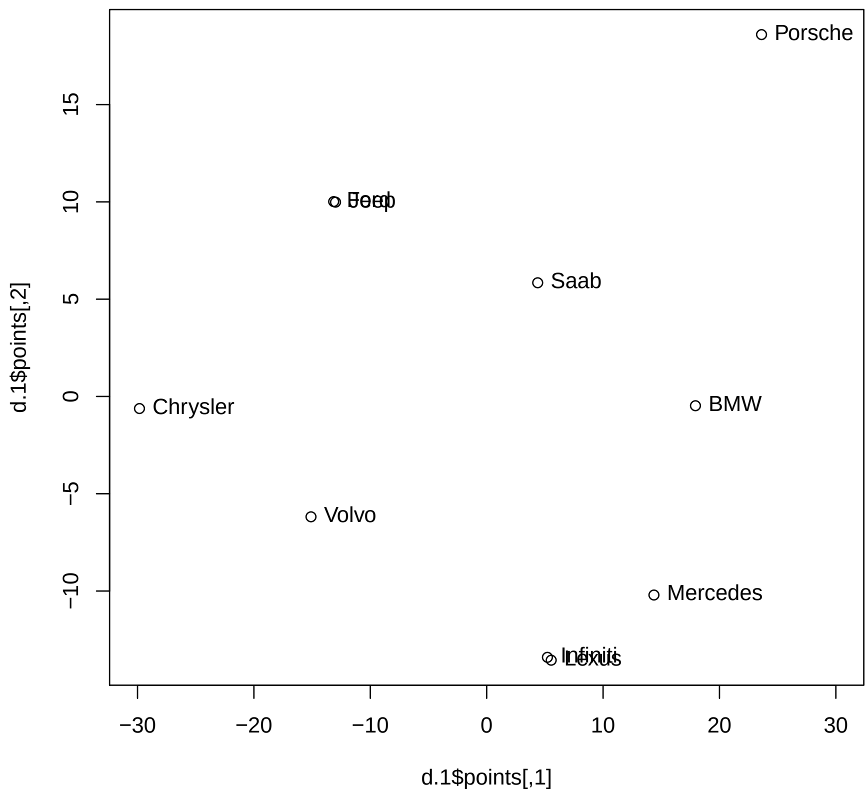

Figure 27: Multidimensional scaling map of cars

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
baseball
```

```
##              TEAM  RS    H  HR   AVG   W  ERA  RA   SO  BB   E
## 1         Toronto 891 1480 232 0.269  93 3.80 670 1117 397  88
## 2      NY Yankees 764 1397 212 0.251  87 4.05 698 1370 474  92
## 3           Texas 751 1419 172 0.257  88 4.24 733 1095 508 119
## 4          Boston 748 1496 161 0.265  78 4.31 753 1218 478  97
## 5        Colorado 737 1479 186 0.265  68 5.04 844 1112 579  95
## 6         Houston 729 1363 230 0.250  86 3.57 618 1280 423  85
## 7     Kansas City 724 1497 139 0.269  95 3.73 641 1160 489  88
## 8         Arizona 720 1494 154 0.264  79 4.04 713 1215 500  86
## 9       Baltimore 713 1370 217 0.250  81 4.05 693 1233 483  77
## 10     Washington 703 1363 177 0.251  83 3.62 635 1342 364  90
## 11     Pittsburgh 697 1462 140 0.260  98 3.21 596 1338 453 122
## 12 San Francisco 696 1486 136 0.267  84 3.72 627 1165 431  78
## 13      Minnesota 696 1349 156 0.247  83 4.07 700 1046 413  86
## 14        Oakland 694 1405 146 0.251  68 4.14 729 1179 474 126
## 15        Detroit 689 1515 151 0.270  74 4.64 803 1100 489  86
## 16   Chicago Cubs 689 1341 171 0.244  97 3.36 608 1431 407 111
## 17        NY Mets 683 1351 177 0.244  90 3.43 613 1337 383  88
## 18      Cleveland 669 1395 141 0.256  81 3.67 640 1407 425  79
## 19     LA Dodgers 667 1346 187 0.250  92 3.44 595 1396 395  75
## 20      LA Angels 661 1331 176 0.246  85 3.94 675 1221 466  93
## 21        Seattle 656 1379 198 0.249  76 4.16 726 1283 491  94
## 22      Milwaukee 655 1378 145 0.251  68 4.28 737 1260 517 116
## 23      San Diego 650 1324 148 0.243  74 4.09 731 1393 516  92
## 24      St. Louis 647 1386 137 0.253 100 2.94 525 1329 477  96
## 25      Tampa Bay 644 1383 167 0.252  80 3.74 642 1355 477  95
## 26     Cincinnati 640 1382 167 0.248  64 4.33 754 1252 544  90
## 27   Philadelphia 626 1374 130 0.249  63 4.69 809 1153 488 117
## 28    Chicago Sox 622 1381 136 0.250  76 3.98 701 1359 474 101
## 29          Miami 613 1420 120 0.260  71 4.02 678 1152 508  77
## 30        Atlanta 573 1361 100 0.251  67 4.41 760 1148 550  90
```

Figure 28: Baseball data

```
baseball.1=princomp(baseball[,-1],cor=T)
summary(baseball.1)

## Importance of components:
##                           Comp.1    Comp.2    Comp.3   Comp.4     Comp.5
## Standard deviation     1.9780942 1.6284703 1.1592964 1.001404 0.72347227
## Proportion of Variance 0.3912857 0.2651916 0.1343968 0.100281 0.05234121
## Cumulative Proportion  0.3912857 0.6564772 0.7908740 0.891155 0.94349624
##                            Comp.6     Comp.7      Comp.8      Comp.9
## Standard deviation     0.58886348 0.33939375 0.269085919 0.158853043
## Proportion of Variance 0.03467602 0.01151881 0.007240723 0.002523429
## Cumulative Proportion  0.97817226 0.98969108 0.996931800 0.999455229
##                             Comp.10
## Standard deviation     0.0738086322
## Proportion of Variance 0.0005447714
## Cumulative Proportion  1.0000000000
```

Figure 29: Principal components analysis of baseball data

```
plot(baseball.1,type="l")
```

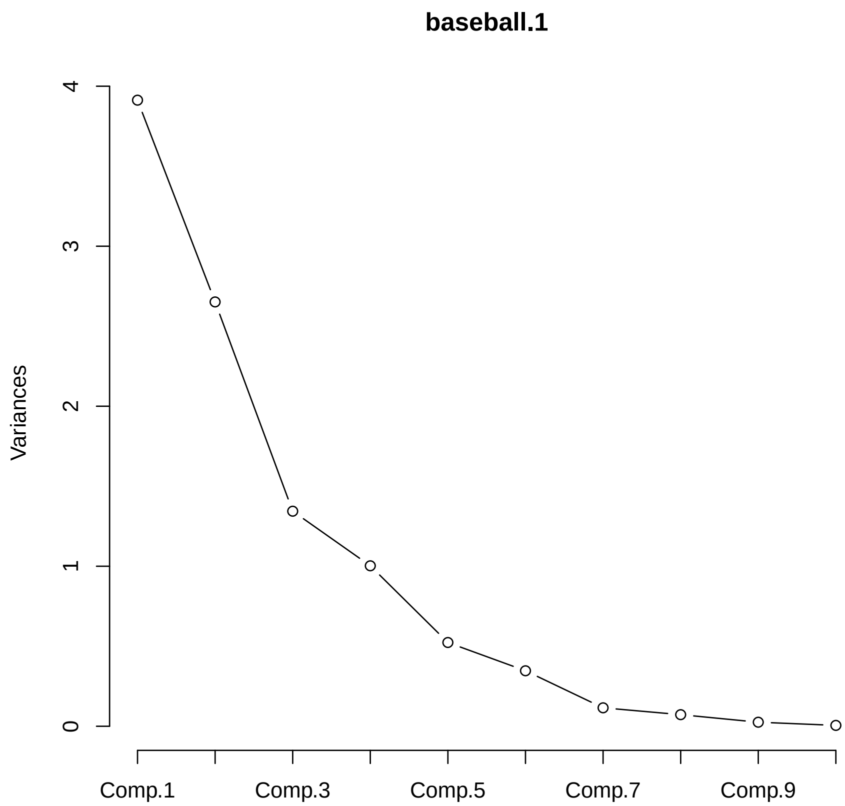

Figure 30: Scree plot of baseball data

```r
baseball.2=factanal(baseball[,-1],4,scores="r")
baseball.2$uniquenesses

##         RS          H         HR        AVG          W        ERA
## 0.00500000 0.03362356 0.24671588 0.00500000 0.16953992 0.00500000
##         RA         SO         BB          E
## 0.00500000 0.47513335 0.41234633 0.48432288

baseball.2$PVAL

## objective
## 0.3670648

baseball.2$loadings

##
## Loadings:
##     Factor1 Factor2 Factor3 Factor4
## RS  -0.131   0.480   0.861
## H    0.113   0.975
## HR          -0.125   0.846  -0.117
## AVG          0.984          -0.141
## W   -0.851   0.134   0.297
## ERA  0.989   0.121
## RA   0.977   0.108           0.171
## SO  -0.538  -0.449  -0.112   0.148
## BB   0.666   0.134  -0.342
## E                            0.707
##
##                Factor1 Factor2 Factor3 Factor4
## SS loadings      3.441   2.434   1.680   0.604
## Proportion Var   0.344   0.243   0.168   0.060
## Cumulative Var   0.344   0.587   0.755   0.816
```

Figure 31: Factor analysis for baseball data

```
biplot(baseball.2$scores,baseball.2$loadings,cex=c(1,0.5))
```

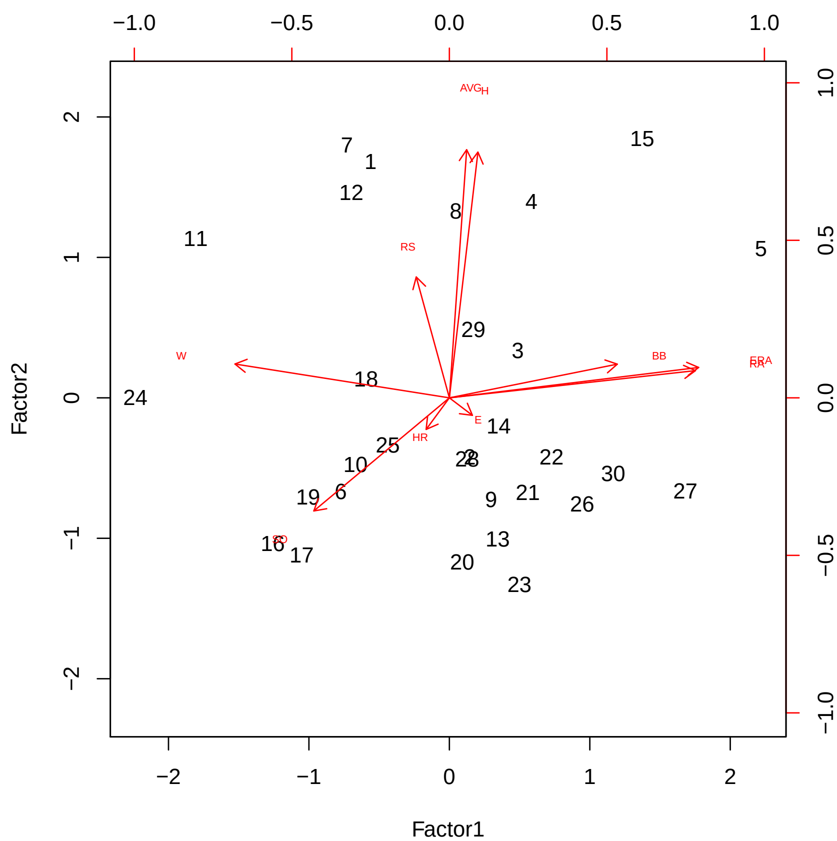

Figure 32: Biplot of baseball factor analysis

```r
genders=c("male","female")
ruptures=c("no","yes")
cesareans=c("no","yes")
induceds=c("no","yes")
combos=expand.grid(induced=induceds,cesarean=cesareans,
                   rupture=ruptures,gender=genders)
freqs=c(177,45,37,18,104,16,9,7,137,53,24,12,74,15,8,2)
births=cbind(combos,freq=freqs)
births
```

```
##    induced cesarean rupture gender freq
## 1       no       no      no   male  177
## 2      yes       no      no   male   45
## 3       no      yes      no   male   37
## 4      yes      yes      no   male   18
## 5       no       no     yes   male  104
## 6      yes       no     yes   male   16
## 7       no      yes     yes   male    9
## 8      yes      yes     yes   male    7
## 9       no       no      no female  137
## 10     yes       no      no female   53
## 11      no      yes      no female   24
## 12     yes      yes      no female   12
## 13      no       no     yes female   74
## 14     yes       no     yes female   15
## 15      no      yes     yes female    8
## 16     yes      yes     yes female    2
```

Figure 33: Birth data

```
births.1=glm(freq~gender*rupture*cesarean*induced,data=births,
             family="poisson")
drop1(births.1,test="Chisq")

## Single term deletions
##
## Model:
## freq ~ gender * rupture * cesarean * induced
##                                 Df Deviance    AIC     LRT Pr(>Chi)
## <none>                              0.00000 113.00
## gender:rupture:cesarean:induced  1  0.83372 111.84 0.83372   0.3612
```

Much analysis follows (not shown).
I ended with:

```
births.10=update(births.9,.~.-gender:induced)
drop1(births.10,test="Chisq")

## Single term deletions
##
## Model:
## freq ~ gender + rupture + cesarean + induced + rupture:cesarean +
##     rupture:induced + cesarean:induced
##                  Df Deviance    AIC     LRT Pr(>Chi)
## <none>                8.9101 105.91
## gender            1  19.4284 114.43 10.5182 0.001182 **
## rupture:cesarean  1  13.9746 108.98  5.0645 0.024421 *
## rupture:induced   1  14.4372 109.44  5.5271 0.018725 *
## cesarean:induced  1  15.9388 110.94  7.0286 0.008022 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 34: Birth data log-linear analysis

```r
xt1=xtabs(freq~rupture+cesarean,data=births)
prop.table(xt1,1)

##        cesarean
## rupture        no       yes
##     no  0.8190855 0.1809145
##     yes 0.8893617 0.1106383

xt2=xtabs(freq~rupture+induced,data=births)
prop.table(xt2,1)

##        induced
## rupture        no       yes
##     no  0.7455268 0.2544732
##     yes 0.8297872 0.1702128

xt3=xtabs(freq~cesarean+induced,data=births)
prop.table(xt3,2)

##         induced
## cesarean        no       yes
##      no  0.8631579 0.7678571
##      yes 0.1368421 0.2321429
```

Figure 35: Birth data sub-tables