Homework 5

```
library(MASS)
library(DAAG)

library(glmnet)

library(FrF2)
```

Question 11.1

Using the crime data set uscrime.txt from Questions 8.2, 9.1, and 10.1, build a regression model

using:

1. Stepwise regression

2. Lasso

3. Elastic net

1. Stepwise regression

```
crime<-
read.table("http://www.statsci.org/data/general/uscrime.txt",header=TRUE)
head(crime)

##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6
##       Prob    Time Crime
## 1 0.084602 26.2011   791
## 2 0.029599 25.2999  1635
## 3 0.083401 24.3006   578
## 4 0.015801 29.9012  1969
## 5 0.041399 21.2998  1234
## 6 0.034201 20.9995   682
```

Scale the predictors except the categorical (So)

```
crimes <- as.data.frame(scale(crime[,c(1,3,4,5,6,7,8,9,10,11,12,13,14,15)]))
crimes <- cbind(crime[,2],crimes,crime[,16]) # Add So Crime back in
```

```
colnames(crimes)[1] <- "So"
colnames(crimes)[16] <- "Crime"
head(crimes)
```

```
##   So          M         Ed        Po1        Po2         LF         M.F
## 1  1  0.9886930 -1.3085099 -0.9085105 -0.8666988 -1.2667456 -1.12060499
## 2  0  0.3521372  0.6580587  0.6056737  0.5280852  0.5396568  0.98341752
## 3  1  0.2725678 -1.4872888 -1.3459415 -1.2958632 -0.6976051 -0.47582390
## 4  0 -0.2048491  1.3731746  2.1535064  2.1732150  0.3911854  0.37257228
## 5  0  0.1929983  1.3731746  0.8075649  0.7426673  0.7376187  0.06714965
## 6  0 -1.3983912  0.3898903  1.1104017  1.2433590 -0.3511718 -0.64550313
##           Pop           NW          U1         U2     Wealth       Ineq
## 1 -0.09500679  1.943738564  0.69510600  0.8313680 -1.3616094  1.6793638
## 2 -0.62033844  0.008483424  0.02950365  0.2393332  0.3276683  0.0000000
## 3 -0.48900552  1.146296747 -0.08143007 -0.1158877 -2.1492481  1.4036474
## 4  3.16204944 -0.205464381  0.36230482  0.5945541  1.5298536 -0.6767585
## 5 -0.48900552 -0.691709391 -0.24783066 -1.6551781  0.5453053 -0.5013026
## 6 -0.30513945 -0.555560788 -0.63609870 -0.5895155  1.6956723 -1.7044289
##         Prob        Time Crime
## 1  1.6497631 -0.05599367   791
## 2 -0.7693365 -0.18315796  1635
## 3  1.5969416 -0.32416470   578
## 4 -1.3761895  0.46611085  1969
## 5 -0.2503580 -0.74759413  1234
## 6 -0.5669349 -0.78996812   682
```

Fit the model with all the predictors

```
lm <- lm(Crime ~ ., data = crime)
summary(lm)
```

```
##
## Call:
## lm.default(formula = Crime ~ ., data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.74  -98.09   -6.69  112.99  512.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            8.783e+01  4.171e+01   2.106 0.043443 *
## So          -3.803e+00  1.488e+02  -0.026 0.979765
## Ed           1.883e+02  6.209e+01   3.033 0.004861 **
## Po1          1.928e+02  1.061e+02   1.817 0.078892 .
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830
## LF          -6.638e+02  1.470e+03  -0.452 0.654654
## M.F          1.741e+01  2.035e+01   0.855 0.398995
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845
## NW           4.204e+00  6.481e+00   0.649 0.521279
```

```
## U1          -5.827e+03  4.210e+03  -1.384 0.176238
## U2           1.678e+02  8.234e+01   2.038 0.050161 .
## Wealth       9.617e-02  1.037e-01   0.928 0.360754
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
## Time        -3.479e+00  7.165e+00  -0.486 0.630708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 209.1 on 31 degrees of freedom
## Multiple R-squared:  0.8031, Adjusted R-squared:  0.7078
## F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

Use stepwise method to re-fit the model with all the predictors

```
stp<-stepAIC(lm, direction="both")
```

```
## Start:  AIC=514.65
## Crime ~ M + So + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 +
##     U2 + Wealth + Ineq + Prob + Time
##
##          Df Sum of Sq     RSS    AIC
## - So      1        29 1354974 512.65
## - LF      1      8917 1363862 512.96
## - Time    1     10304 1365250 513.00
## - Pop     1     14122 1369068 513.14
## - NW      1     18395 1373341 513.28
## - M.F     1     31967 1386913 513.74
## - Wealth  1     37613 1392558 513.94
## - Po2     1     37919 1392865 513.95
## <none>                1354946 514.65
## - U1      1     83722 1438668 515.47
## - Po1     1    144306 1499252 517.41
## - U2      1    181536 1536482 518.56
## - M       1    193770 1548716 518.93
## - Prob    1    199538 1554484 519.11
## - Ed      1    402117 1757063 524.86
## - Ineq    1    423031 1777977 525.42
##
## Step:  AIC=512.65
## Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +
##     Wealth + Ineq + Prob + Time
##
##          Df Sum of Sq     RSS    AIC
## - Time    1     10341 1365315 511.01
## - LF      1     10878 1365852 511.03
## - Pop     1     14127 1369101 511.14
## - NW      1     21626 1376600 511.39
## - M.F     1     32449 1387423 511.76
## - Po2     1     37954 1392929 511.95
```

```
## - Wealth  1     39223 1394197 511.99
## <none>                1354974 512.65
## - U1      1     96420 1451395 513.88
## + So      1        29 1354946 514.65
## - Po1     1    144302 1499277 515.41
## - U2      1    189859 1544834 516.81
## - M       1    195084 1550059 516.97
## - Prob    1    204463 1559437 517.26
## - Ed      1    403140 1758114 522.89
## - Ineq    1    488834 1843808 525.13
##
## Step:  AIC=511.01
## Crime ~ M + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 + U2 +
##     Wealth + Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - LF      1     10533 1375848 509.37
## - NW      1     15482 1380797 509.54
## - Pop     1     21846 1387161 509.75
## - Po2     1     28932 1394247 509.99
## - Wealth  1     36070 1401385 510.23
## - M.F     1     41784 1407099 510.42
## <none>                1365315 511.01
## - U1      1     91420 1456735 512.05
## + Time    1     10341 1354974 512.65
## + So      1        65 1365250 513.00
## - Po1     1    134137 1499452 513.41
## - U2      1    184143 1549458 514.95
## - M       1    186110 1551425 515.01
## - Prob    1    237493 1602808 516.54
## - Ed      1    409448 1774763 521.33
## - Ineq    1    502909 1868224 523.75
##
## Step:  AIC=509.37
## Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + NW + U1 + U2 + Wealth +
##     Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - NW      1     11675 1387523 507.77
## - Po2     1     21418 1397266 508.09
## - Pop     1     27803 1403651 508.31
## - M.F     1     31252 1407100 508.42
## - Wealth  1     35035 1410883 508.55
## <none>                1375848 509.37
## - U1      1     80954 1456802 510.06
## + LF      1     10533 1365315 511.01
## + Time    1      9996 1365852 511.03
## + So      1      3046 1372802 511.26
## - Po1     1    123896 1499744 511.42
## - U2      1    190746 1566594 513.47
```

```
## - M       1    217716 1593564 514.27
## - Prob    1    226971 1602819 514.54
## - Ed      1    413254 1789103 519.71
## - Ineq    1    500944 1876792 521.96
##
## Step:  AIC=507.77
## Crime ~ M + Ed + Po1 + Po2 + M.F + Pop + U1 + U2 + Wealth + Ineq +
##     Prob
##
##          Df Sum of Sq     RSS    AIC
## - Po2     1     16706 1404229 506.33
## - Pop     1     25793 1413315 506.63
## - M.F     1     26785 1414308 506.66
## - Wealth  1     31551 1419073 506.82
## <none>                1387523 507.77
## - U1      1     83881 1471404 508.52
## + NW      1     11675 1375848 509.37
## + So      1      7207 1380316 509.52
## + LF      1      6726 1380797 509.54
## + Time    1      4534 1382989 509.61
## - Po1     1    118348 1505871 509.61
## - U2      1    201453 1588976 512.14
## - Prob    1    216760 1604282 512.59
## - M       1    309214 1696737 515.22
## - Ed      1    402754 1790276 517.74
## - Ineq    1    589736 1977259 522.41
##
## Step:  AIC=506.33
## Crime ~ M + Ed + Po1 + M.F + Pop + U1 + U2 + Wealth + Ineq +
##     Prob
##
##          Df Sum of Sq     RSS    AIC
## - Pop     1     22345 1426575 505.07
## - Wealth  1     32142 1436371 505.39
## - M.F     1     36808 1441037 505.54
## <none>                1404229 506.33
## - U1      1     86373 1490602 507.13
## + Po2     1     16706 1387523 507.77
## + NW      1      6963 1397266 508.09
## + So      1      3807 1400422 508.20
## + LF      1      1986 1402243 508.26
## + Time    1       575 1403654 508.31
## - U2      1    205814 1610043 510.76
## - Prob    1    218607 1622836 511.13
## - M       1    307001 1711230 513.62
## - Ed      1    389502 1793731 515.83
## - Ineq    1    608627 2012856 521.25
## - Po1     1   1050202 2454432 530.57
##
## Step:  AIC=505.07
```

```
## Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Wealth + Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## - Wealth  1     26493 1453068 503.93
## <none>                1426575 505.07
## - M.F     1     84491 1511065 505.77
## - U1      1     99463 1526037 506.24
## + Pop     1     22345 1404229 506.33
## + Po2     1     13259 1413315 506.63
## + NW      1      5927 1420648 506.87
## + So      1      5724 1420851 506.88
## + LF      1      5176 1421398 506.90
## + Time    1      3913 1422661 506.94
## - Prob    1    198571 1625145 509.20
## - U2      1    208880 1635455 509.49
## - M       1    320926 1747501 512.61
## - Ed      1    386773 1813348 514.35
## - Ineq    1    594779 2021354 519.45
## - Po1     1   1127277 2553852 530.44
##
## Step:  AIC=503.93
## Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob
##
##          Df Sum of Sq     RSS    AIC
## <none>                1453068 503.93
## + Wealth  1     26493 1426575 505.07
## - M.F     1    103159 1556227 505.16
## + Pop     1     16697 1436371 505.39
## + Po2     1     14148 1438919 505.47
## + So      1      9329 1443739 505.63
## + LF      1      4374 1448694 505.79
## + NW      1      3799 1449269 505.81
## + Time    1      2293 1450775 505.86
## - U1      1    127044 1580112 505.87
## - Prob    1    247978 1701046 509.34
## - U2      1    255443 1708511 509.55
## - M       1    296790 1749858 510.67
## - Ed      1    445788 1898855 514.51
## - Ineq    1    738244 2191312 521.24
## - Po1     1   1672038 3125105 537.93
```

See what does the final model looks like

```
stp$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## Crime ~ M + So + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 +
```

```
##     U2 + Wealth + Ineq + Prob + Time
##
## Final Model:
## Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob
##
##
##       Step Df    Deviance Resid. Df Resid. Dev      AIC
## 1                                31    1354946 514.6488
## 2     - So  1    28.57405        32    1354974 512.6498
## 3   - Time  1 10340.66984        33    1365315 511.0072
## 4     - LF  1 10533.15902        34    1375848 509.3684
## 5     - NW  1 11674.63991        35    1387523 507.7655
## 6    - Po2  1 16706.34095        36    1404229 506.3280
## 7    - Pop  1 22345.36638        37    1426575 505.0700
## 8 - Wealth  1 26493.24677        38    1453068 503.9349
```

Get the coefficients of the predictors

```
summary(stp)
```

```
##
## Call:
## lm.default(formula = Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq +
##     Prob, data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -444.70 -111.07    3.03  122.15  483.30
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6426.10    1194.61  -5.379 4.04e-06 ***
## M              93.32      33.50   2.786  0.00828 **
## Ed            180.12      52.75   3.414  0.00153 **
## Po1           102.65      15.52   6.613 8.26e-08 ***
## M.F            22.34      13.60   1.642  0.10874
## U1          -6086.63    3339.27  -1.823  0.07622 .
## U2            187.35      72.48   2.585  0.01371 *
## Ineq           61.33      13.96   4.394 8.63e-05 ***
## Prob        -3796.03    1490.65  -2.547  0.01505 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 195.5 on 38 degrees of freedom
## Multiple R-squared:  0.7888, Adjusted R-squared:  0.7444
## F-statistic: 17.74 on 8 and 38 DF,  p-value: 1.159e-10
```

Cross validate the model.Since we only have 47 data points, we will use 47-fold cross-validation (leave-one-out cross-validation).

```
sst <- sum((crime$Crime - mean(crime$Crime))^2)
sse <- 0
for(i in 1:nrow(crimes)) {
        step_i = lm(Crime ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob, data
= crimes[-i,])
        pred_i <- predict(step_i,newdata=crimes[i,])
        sse <- sse + ((pred_i - crime[i,16])^2)
}
R2 <- 1 - sse/sst
R2

##         1
## 0.667621
```

For the scaled data, the fitted model is Crime=-6426.1+93.3*M+180.1*Ed+102.7*Po1+22.3*M.F-6086.6*U1+187.3*U2+61.3*Ineq-3796.0*Prob The adjusted R square is 0.744. After cross validation, it is 0.668

2. Lasso

```
set.seed(1234)
x<-as.matrix(crimes[,-16])
y<-as.matrix(crimes$Crime)
lasso<-cv.glmnet(x=x,y=y,alpha=1,
                nfolds = 5,type.measure="mse",family="gaussian")
```

Output the coefficients of the variables selected by lasso

```
coef(lasso, s=lasso$lambda.min)

## 16 x 1 sparse Matrix of class "dgCMatrix"
##                       1
## (Intercept) 891.753091
## So           39.162794
## M            96.857033
## Ed          155.983201
## Po1         299.176316
## Po2           .
## LF            .
## M.F          55.298574
## Pop          -5.983451
## NW            9.847441
## U1          -53.682330
## U2           93.426615
## Wealth       27.693936
## Ineq        218.415654
```

```
## Prob        -85.785382
## Time          .
```

Fit a new model with these 11 variables

```
lasso2<-lm(Crime ~So+M+Ed+Po1+LF+M.F+NW+U1+U2+Ineq+Prob, data = crimes)
summary(lasso2)

##
## Call:
## lm.default(formula = Crime ~ So + M + Ed + Po1 + LF + M.F + NW +
##     U1 + U2 + Ineq + Prob, data = crimes)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -443.2 -101.4    4.1  120.5  486.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   892.63      55.99  15.943  < 2e-16 ***
## So             36.57     139.62   0.262  0.79489
## M             106.61      49.29   2.163  0.03747 *
## Ed            209.15      65.00   3.218  0.00278 **
## Po1           295.60      54.50   5.424 4.44e-06 ***
## LF            -10.69      54.11  -0.198  0.84447
## M.F            74.96      51.13   1.466  0.15159
## NW             13.01      59.46   0.219  0.82814
## U1           -109.08      71.71  -1.521  0.13725
## U2            151.47      65.99   2.295  0.02783 *
## Ineq          233.00      67.67   3.443  0.00151 **
## Prob          -96.00      39.58  -2.425  0.02059 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 202.9 on 35 degrees of freedom
## Multiple R-squared:  0.7906, Adjusted R-squared:  0.7248
## F-statistic: 12.01 on 11 and 35 DF,  p-value: 6.965e-09
```

The adjusted R square is 0.725, a little bit lower than the previous model. However, hard to tell which one is better.

Same as the previous one, cross validate the model using 47-fold cross-validation.

```
sse2 <- 0
for(i in 1:nrow(crimes)) {
        step_i <- lm(Crime ~ So+M+Ed+Po1+LF+M.F+NW+U1+U2+Ineq+Prob, data =
crimes[-i,])
```

```
        pred_i <- predict(step_i,newdata=crimes[i,])
        sse2 <- sse2 + ((pred_i - crime[i,16])^2)
}
R2.2 <- 1 - sse2/sst
R2.2

##         1
## 0.5962457
```

The cross validated R square is 0.596, much lower than the previous model (0.668),I would prefer the first model, which also have fewer predictors.

3. Elastic net #Use alpha from 0 to 1, by 0.1, and calculate the R-Squared values

```
set.seed(123)
R2.3<-c()
for (i in 0:10) {
        elastic<-cv.glmnet(x=x,y=y,alpha=i/10,nfolds =
5,type.measure="mse",family="gaussian")

        R2.3<-
cbind(R2.3,elastic$glmnet.fit$dev.ratio[which(elastic$glmnet.fit$lambda ==
elastic$lambda.min)])

}

R2.3

##           [,1]      [,2]     [,3]      [,4]      [,5]      [,6]      [,7]
## [1,] 0.7305131 0.7441572 0.769993 0.3977604 0.7276302 0.7552074 0.7043895
##           [,8]      [,9]     [,10]     [,11]
## [1,] 0.7844037 0.7483184 0.7365694 0.6373179
```

Get the best value of alpha

```
alpha_best <-(which.max(R2.3)-1)/10
alpha_best

## [1] 0.7
```

Re-fit the model using this alpha value.

```
set.seed(123)
elastic2<-cv.glmnet(x=x,y=y,alpha=alpha_best,nfolds =
5,type.measure="mse",family="gaussian")
coef(elastic2, s=elastic2$lambda.min)
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                     1
## (Intercept) 893.35958
## So           34.44373
## M           103.20631
## Ed          173.14284
## Po1         292.15108
## Po2           .
## LF           -17.86024
## M.F          54.43589
## Pop           .
## NW           16.60652
## U1          -72.19388
## U2          116.46339
## Wealth        .
## Ineq        245.02692
## Prob        -89.98027
## Time          .
```

Fit a new model with these 11 variables

```
elastic3<-lm(Crime ~So+M+Ed+Po1+LF+MF+NW+U1+U2+Ineq+Prob, data = crimes)
summary(elastic3)
```

```
##
## Call:
## lm.default(formula = Crime ~ So + M + Ed + Po1 + LF + M.F + NW +
##     U1 + U2 + Ineq + Prob, data = crimes)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -443.2 -101.4    4.1  120.5  486.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   892.63      55.99  15.943  < 2e-16 ***
## So             36.57     139.62   0.262  0.79489
## M             106.61      49.29   2.163  0.03747 *
## Ed            209.15      65.00   3.218  0.00278 **
## Po1           295.60      54.50   5.424 4.44e-06 ***
## LF            -10.69      54.11  -0.198  0.84447
## M.F            74.96      51.13   1.466  0.15159
## NW             13.01      59.46   0.219  0.82814
## U1           -109.08      71.71  -1.521  0.13725
## U2            151.47      65.99   2.295  0.02783 *
## Ineq          233.00      67.67   3.443  0.00151 **
## Prob          -96.00      39.58  -2.425  0.02059 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 202.9 on 35 degrees of freedom
## Multiple R-squared:  0.7906, Adjusted R-squared:  0.7248
## F-statistic: 12.01 on 11 and 35 DF,  p-value: 6.965e-09
```

The adjusted R square is 0.725, same to the Lasso model.

Now do a cross validation on the model using 47-fold cross-validation.

```
sse2.3 <- 0
for(i in 1:nrow(crimes)) {
        step_i <- lm(Crime ~
So+M+Ed+Po1+Po2+LF+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob, data = crimes[-i,])
        pred_i <- predict(step_i,newdata=crimes[i,])
        sse2.3 <- sse2.3 + ((pred_i - crime[i,16])^2)
}
R2.3b <- 1 - sse2.3/sst
R2.3b

##         1
## 0.5361309
```

The cross-validated R square is 0.536. This is the lowest R square among the 3-models.In conclusion, the first model generated by stepwise method is the best.

Question 12.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a design of #experiments approach would be appropriate.

The bubble milk teahouse would like to attract more people to buy their new arrival milk tea.They need to decide what cup design works best: #including the combination of font, size, picture of the milk / tea / bubble / other ingredients.

Question 12.2

To determine the value of 10 different yes/no features to the market value of a house (large yard, solar #roof, etc.), a real estate agent plans to survey 50 potential buyers, showing a fictitious house with #different combinations of features. To reduce the survey size, the agent wants to show just 16 fictitious houses.

Generate a 16 run, 10 factors, 2 levels (yes/no) fractional factorial design

```
set.seed(123)
design<-FrF2(nruns = 16,nfactors = 10)
design

##     A  B  C  D  E  F  G  H  J  K
## 1  -1 -1  1 -1  1 -1 -1  1  1 -1
## 2   1  1 -1  1  1 -1 -1  1 -1 -1
## 3   1 -1  1 -1 -1  1 -1 -1  1  1
## 4  -1  1  1  1 -1 -1  1 -1  1 -1
## 5  -1 -1  1  1  1 -1 -1 -1 -1  1
## 6  -1 -1 -1 -1  1  1  1  1 -1  1
## 7   1 -1  1  1 -1  1 -1  1 -1 -1
## 8  -1 -1 -1  1  1  1  1 -1  1 -1
## 9   1  1  1  1  1  1  1  1  1  1
## 10  1  1 -1 -1  1 -1 -1 -1  1  1
## 11  1 -1 -1  1 -1 -1  1  1  1  1
## 12 -1  1 -1 -1 -1  1 -1  1  1 -1
## 13  1  1  1 -1  1  1  1 -1 -1 -1
## 14  1 -1 -1 -1 -1 -1  1 -1 -1 -1
## 15 -1  1 -1  1 -1  1 -1 -1 -1  1
## 16 -1  1  1 -1 -1 -1  1  1 -1  1
## class=design, type= FrF2
```

Question 13.1

For each of the following distributions, give an example of data that you would expect to follow this distribution (besides the examples already discussed in class).

a. **Binomial**

   The probability to win a game is p, to lose is 1-p. The probability of getting x wins out of n games follows Binomial distribution.

b. **Geometric**

   Same game to the previous one. The probability of have x wins until the first failure follows Geometric distribution.

c. **Poisson**

   The average number of customers arrived at a Starbucks per hour, follows Poisson distribution.

d. **Exponential**

   When the average number of customers arrived at a Starbucks per hour follows Poisson distribution, the time between successive arrivals follows Exponential distribution.

e. **Weibull**

   The lifetime of the credit card magnetic strip follows Weibull distribution.