# Week 4 Homework

*Matthew Dunne*

*April 17, 2018*

## 1 Data

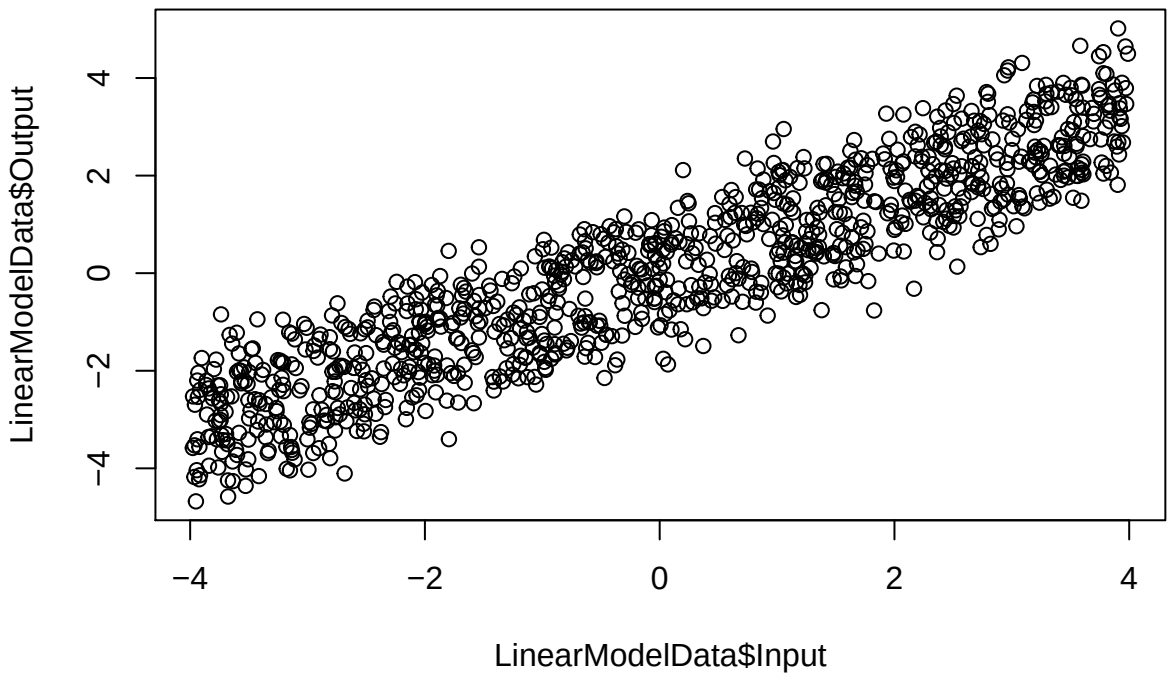
Look at the sample in the file ResidualAnalysisProjectData_1.csv: The first rows and the X-Y plot are:

```
dataPath<-"C:/Users/mjdun/Desktop/Master Classes/Q1/Statistical Analysis/Lecture 4"
LinearModelData<-read.csv(file=paste(dataPath,"Residual Analysis Project Data_1.csv",sep="/"))
head(LinearModelData)
```

```
##        Input    Output
## 1  3.6664327  2.747905
## 2 -2.5194424 -3.242035
## 3  0.6475581  1.559734
## 4  2.4439621  1.292082
## 5  1.9921334  1.958417
## 6  1.7534556  2.049381
```

and plot the data

```
plot(LinearModelData$Input,LinearModelData$Output)
```



## 2 Fitting linear model

Estimate linear model using function lm() look at the output of the function

```
Estimated.LinearModel <- lm(Output ~ Input,data=LinearModelData)
names(Estimated.LinearModel)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

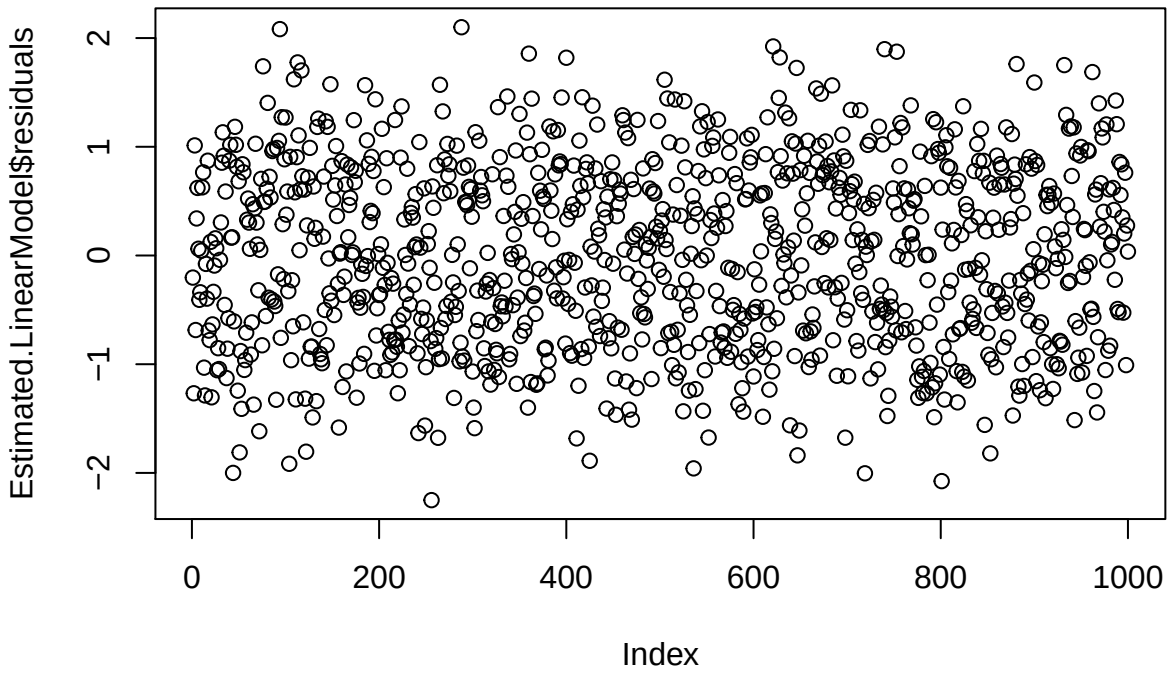
### 2.1 Object lm()

Explore the elements of the object lm: 1. Coefficients
2. Residuals (make a plot). How residuals are calculated?
3. Find out what are fitted.values

```
Estimated.LinearModel$coefficients
```

```
## (Intercept)       Input
##  0.03160231  0.79627673
```

```
plot(Estimated.LinearModel$residuals)
```




### 2.2 Objects of Summary

Look at the summary

```
summary(Estimated.LinearModel)
```

```
##
## Call:
## lm(formula = Output ~ Input, data = LinearModelData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.25025 -0.68362  0.01354  0.66505  2.09946
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03160    0.02653   1.191    0.234
## Input        0.79628    0.01138  69.993   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8389 on 998 degrees of freedom
## Multiple R-squared:  0.8308, Adjusted R-squared:  0.8306
## F-statistic:  4899 on 1 and 998 DF,  p-value: < 2.2e-16
```

Interpret the summary

```
names(summary(Estimated.LinearModel))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

What is summary(Estimated.LinearModel)$sigma?

```
summary(Estimated.LinearModel)$sigma
```

```
## [1] 0.838893
```

```
summary(Estimated.LinearModel)$sigma^2
```

```
## [1] 0.7037415
```

Check how summary(Estimated.LinearModel)$sigma is calculated in the object summary(Estimated.LinearModel) by reproducing the square of it: 1. Using var() (the resulting variable is sigmaSquared.byVar) 2. Using only sum() (the resulting variable is sigmaSquared.bySum)

```
sigmaSquared.byVar<-var(Estimated.LinearModel$residuals)*999/998
sigmaSquared.bySum<-sum((Estimated.LinearModel$residuals-mean(Estimated.LinearModel$residuals))^2)/998
c(sigmaSquared.byVar=sigmaSquared.byVar,sigmaSquared.bySum=sigmaSquared.bySum,fromModel=summary(Estimat
```

```
## sigmaSquared.byVar sigmaSquared.bySum          fromModel
##          0.7037415          0.7037415          0.7037415
```

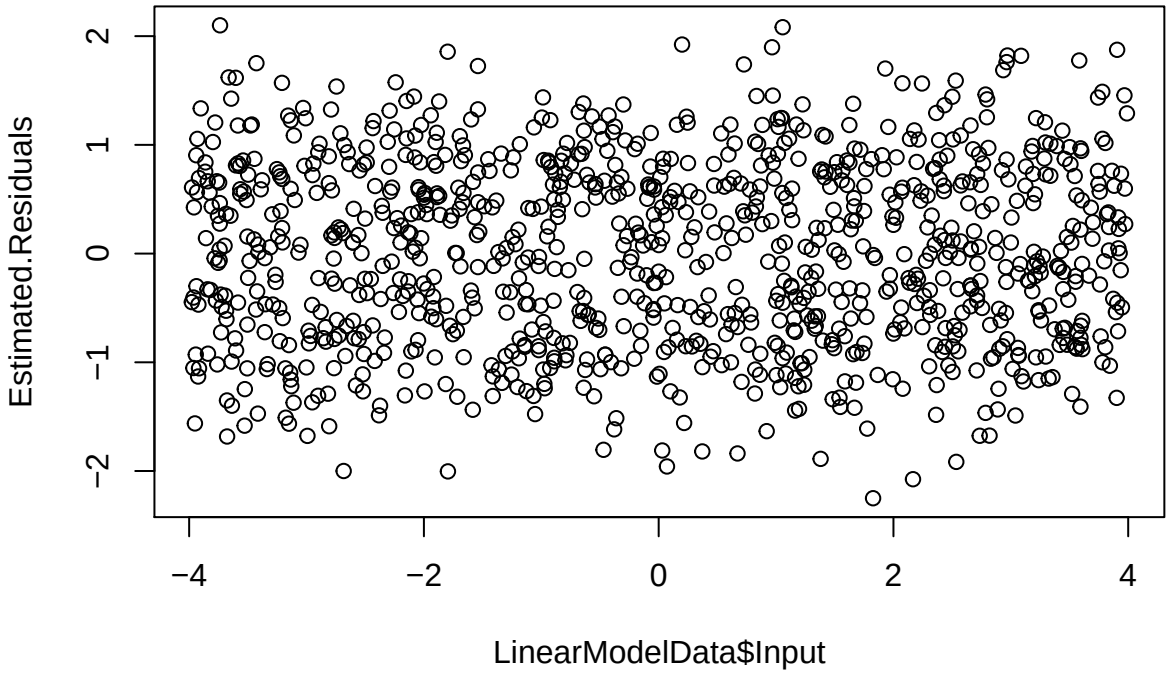
When we compare the two calculations with summary(Estimated.LinearModel)$sigma^2 we see they are the same. ***The var() function uses n-1. The model uses (sum of (residuals^2))/n-2. Multiply sigmaSquared.by.Var by* 999/998 to "reconcile it".***

## 3. Analysis Residuals

### 3.1 Residuals of the Model

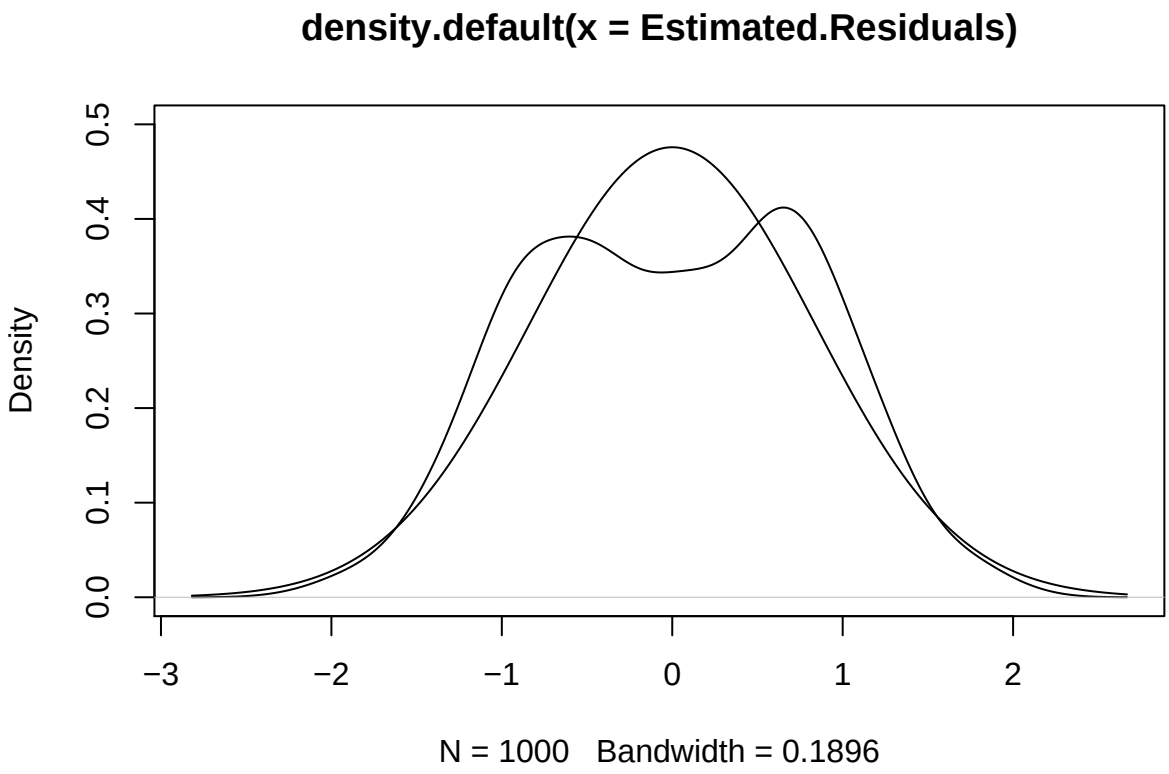
Observe the residuals, plot them against the input.

```
Estimated.Residuals <- Estimated.LinearModel$residuals
plot(LinearModelData$Input, Estimated.Residuals)
```




, and their probability density in comparison with the normal density

```
#plot residuals
Probability.Density.Residuals <- density(Estimated.Residuals)
plot(Probability.Density.Residuals, ylim = c(0, 0.5))
#add line of what distribution would look like if residuals were normally distributed
lines(Probability.Density.Residuals$x, dnorm(Probability.Density.Residuals$x,
    mean = mean(Estimated.Residuals), sd = sd(Estimated.Residuals)))
```




***What do you conclude from the analysis of residuals?*** The residuals are not normally distributed. There is a bi-modal distribution.

### 3.2 Clustering Sample

Calculate mean values of negative residuals and positive residuals.

```
c(Left.Mean = mean(Estimated.Residuals[Estimated.Residuals < 0]),
  Right.Mean = mean(Estimated.Residuals[Estimated.Residuals > 0]))
```

```
##  Left.Mean Right.Mean
## -0.7241664  0.7013580
```

Separate the given sample into 2 subsamples: one, for which the residuals are below zero and another, for which they are above zero. Create variable Unscrambled.Selection.Sequence estimating switching between the two subsamples (1 corresponds to the positive residual case and 0 corresponds to the negative residual case).

```
head(Estimated.Residuals)
```

```
##          1          2          3          4          5          6
## -0.2031922 -1.2674642  1.0124960 -0.6855901  0.3405253  0.6215430
```

```
Unscrambled.Selection.Sequence<-transform(Estimated.Residuals, Estimated.Residuals = ifelse(Estimated.R
#Unscrambled.Selection.Sequence= ifelse(Estimated.Residuals < 0, 0, 1)
#Ahbi said a better way should be:
#fisrt assign a value(0)to the residuals, and R will return the vector of "Trues" and "Falses"
Estimated.Residuals < 0
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12
##  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE
##    13    14    15    16    17    18    19    20    21    22    23    24
##  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
##    25    26    27    28    29    30    31    32    33    34    35    36
## FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
##    37    38    39    40    41    42    43    44    45    46    47    48
##  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE
##    49    50    51    52    53    54    55    56    57    58    59    60
##  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE
##    61    62    63    64    65    66    67    68    69    70    71    72
## FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE
##    73    74    75    76    77    78    79    80    81    82    83    84
## FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
##    85    86    87    88    89    90    91    92    93    94    95    96
##  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
##    97    98    99   100   101   102   103   104   105   106   107   108
## FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
##   109   110   111   112   113   114   115   116   117   118   119   120
## FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##   121   122   123   124   125   126   127   128   129   130   131   132
##  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE
##   133   134   135   136   137   138   139   140   141   142   143   144
##  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE
##   145   146   147   148   149   150   151   152   153   154   155   156
## FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE
##   157   158   159   160   161   162   163   164   165   166   167   168
##  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
##   169   170   171   172   173   174   175   176   177   178   179   180
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
##   181   182   183   184   185   186   187   188   189   190   191   192
##  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE
##   193   194   195   196   197   198   199   200   201   202   203   204
## FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE
##   205   206   207   208   209   210   211   212   213   214   215   216
## FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##   217   218   219   220   221   222   223   224   225   226   227   228
## FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE
##   229   230   231   232   233   234   235   236   237   238   239   240
##  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##   241   242   243   244   245   246   247   248   249   250   251   252
##  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
##   253   254   255   256   257   258   259   260   261   262   263   264
## FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
##   265   266   267   268   269   270   271   272   273   274   275   276
## FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
##   277   278   279   280   281   282   283   284   285   286   287   288
##  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE
##   289   290   291   292   293   294   295   296   297   298   299   300
##  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE
##   301   302   303   304   305   306   307   308   309   310   311   312
##  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
##   313   314   315   316   317   318   319   320   321   322   323   324
##  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
```

```
##   325   326   327   328   329   330   331   332   333   334   335   336
##  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
##   337   338   339   340   341   342   343   344   345   346   347   348
## FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE
##   349   350   351   352   353   354   355   356   357   358   359   360
## FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
##   361   362   363   364   365   366   367   368   369   370   371   372
## FALSE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
##   373   374   375   376   377   378   379   380   381   382   383   384
## FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE
##   385   386   387   388   389   390   391   392   393   394   395   396
## FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE
##   397   398   399   400   401   402   403   404   405   406   407   408
##  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE
##   409   410   411   412   413   414   415   416   417   418   419   420
##  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE
##   421   422   423   424   425   426   427   428   429   430   431   432
## FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE
##   433   434   435   436   437   438   439   440   441   442   443   444
## FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE
##   445   446   447   448   449   450   451   452   453   454   455   456
##  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE
##   457   458   459   460   461   462   463   464   465   466   467   468
## FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
##   469   470   471   472   473   474   475   476   477   478   479   480
## FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
##   481   482   483   484   485   486   487   488   489   490   491   492
##  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE
##   493   494   495   496   497   498   499   500   501   502   503   504
## FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE
##   505   506   507   508   509   510   511   512   513   514   515   516
## FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE
##   517   518   519   520   521   522   523   524   525   526   527   528
##  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE
##   529   530   531   532   533   534   535   536   537   538   539   540
##  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE
##   541   542   543   544   545   546   547   548   549   550   551   552
## FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE
##   553   554   555   556   557   558   559   560   561   562   563   564
##  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE
##   565   566   567   568   569   570   571   572   573   574   575   576
##  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE
##   577   578   579   580   581   582   583   584   585   586   587   588
##  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##   589   590   591   592   593   594   595   596   597   598   599   600
##  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE
##   601   602   603   604   605   606   607   608   609   610   611   612
##  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE FALSE
##   613   614   615   616   617   618   619   620   621   622   623   624
## FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
##   625   626   627   628   629   630   631   632   633   634   635   636
##  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE
##   637   638   639   640   641   642   643   644   645   646   647   648
## FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE  TRUE
```

```
##   649   650   651   652   653   654   655   656   657   658   659   660
##  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
##   661   662   663   664   665   666   667   668   669   670   671   672
##  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##   673   674   675   676   677   678   679   680   681   682   683   684
## FALSE FALSE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
##   685   686   687   688   689   690   691   692   693   694   695   696
##  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE
##   697   698   699   700   701   702   703   704   705   706   707   708
##  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##   709   710   711   712   713   714   715   716   717   718   719   720
##  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
##   721   722   723   724   725   726   727   728   729   730   731   732
## FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE
##   733   734   735   736   737   738   739   740   741   742   743   744
##  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
##   745   746   747   748   749   750   751   752   753   754   755   756
##  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
##   757   758   759   760   761   762   763   764   765   766   767   768
## FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
##   769   770   771   772   773   774   775   776   777   778   779   780
## FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE
##   781   782   783   784   785   786   787   788   789   790   791   792
##  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE
##   793   794   795   796   797   798   799   800   801   802   803   804
##  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE
##   805   806   807   808   809   810   811   812   813   814   815   816
## FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE
##   817   818   819   820   821   822   823   824   825   826   827   828
##  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE
##   829   830   831   832   833   834   835   836   837   838   839   840
##  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE FALSE
##   841   842   843   844   845   846   847   848   849   850   851   852
## FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
##   853   854   855   856   857   858   859   860   861   862   863   864
##  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
##   865   866   867   868   869   870   871   872   873   874   875   876
## FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE
##   877   878   879   880   881   882   883   884   885   886   887   888
##  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
##   889   890   891   892   893   894   895   896   897   898   899   900
##  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE
##   901   902   903   904   905   906   907   908   909   910   911   912
## FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
##   913   914   915   916   917   918   919   920   921   922   923   924
## FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
##   925   926   927   928   929   930   931   932   933   934   935   936
##  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE
##   937   938   939   940   941   942   943   944   945   946   947   948
## FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
##   949   950   951   952   953   954   955   956   957   958   959   960
## FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
##   961   962   963   964   965   966   967   968   969   970   971   972
##  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE
```

```
##   973   974   975   976   977   978   979   980   981   982   983   984
## FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE
##   985   986   987   988   989   990   991   992   993   994   995   996
## FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
##   997   998   999  1000
## FALSE  TRUE FALSE FALSE
```

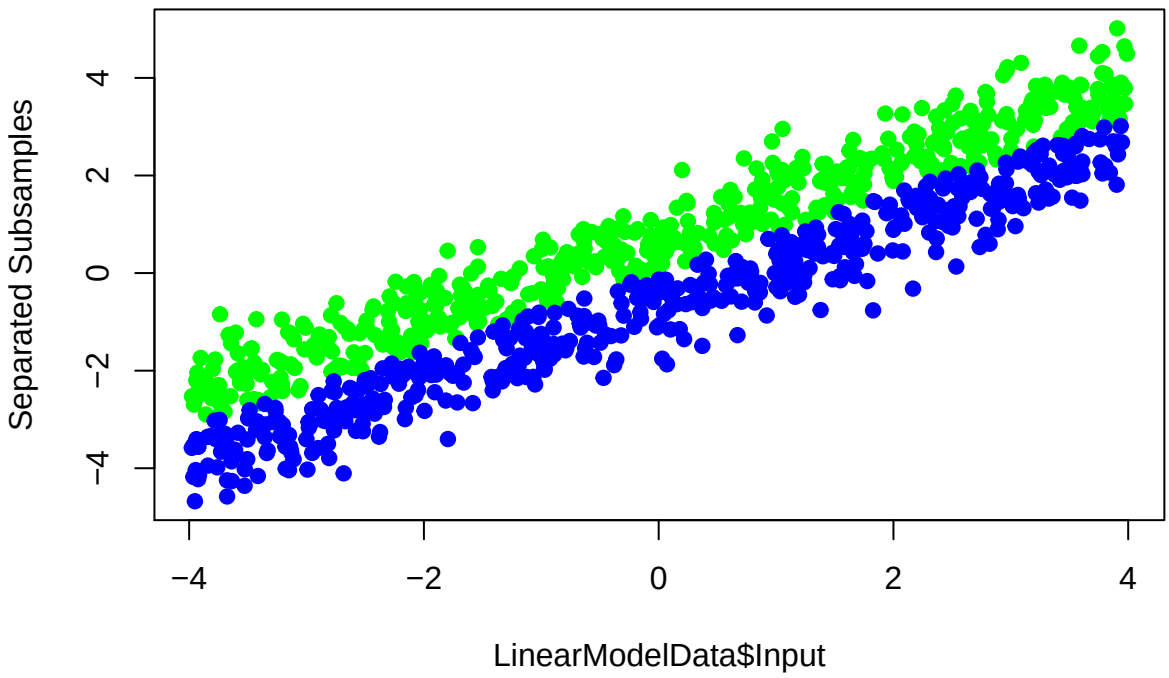
```
#then use the as.numeric function to turn these true and false into numbers
df1<-cbind(LinearModelData, Res=Unscrambled.Selection.Sequence)
df1[df1$Res==0, c("Input", "Output")]<-NA
df2<-cbind(LinearModelData, Res=Unscrambled.Selection.Sequence)
df2[df2$Res==1, c("Input", "Output")]<-NA

LinearModel1.Recovered<-cbind(df1[,1],df1[,2])
LinearModel2.Recovered<-cbind(df2[,1],df2[,2])
head(cbind(LinearModel1.Recovered,LinearModel2.Recovered),30)
```

```
##             [,1]      [,2]       [,3]        [,4]
##  [1,]         NA        NA  3.6664327  2.74790517
##  [2,]         NA        NA -2.5194424 -3.24203530
##  [3,]  0.6475581  1.559734         NA          NA
##  [4,]         NA        NA  2.4439621  1.29208230
##  [5,]  1.9921334  1.958417         NA          NA
##  [6,]  1.7534556  2.049381         NA          NA
##  [7,]  2.7300053  2.267323         NA          NA
##  [8,]         NA        NA  1.2366129  0.60842281
##  [9,]         NA        NA  1.7351840  1.07506483
## [10,]  2.6600869  2.193584         NA          NA
## [11,]  2.5722176  2.706334         NA          NA
## [12,]  1.5666576  2.043858         NA          NA
## [13,]         NA        NA  3.8438847  2.06073032
## [14,]         NA        NA  0.8196281 -0.60317801
## [15,]         NA        NA  0.4030093  0.27503423
## [16,]         NA        NA -0.3165287 -0.61742911
## [17,]  1.1915423  1.852328         NA          NA
## [18,]         NA        NA  3.4420387  2.08164193
## [19,]         NA        NA -2.8572507 -3.02171399
## [20,]  1.3629237  1.242134         NA          NA
## [21,]         NA        NA -2.1617141 -2.99443098
## [22,]         NA        NA  0.7875045  0.02474258
## [23,]         NA        NA -3.8181210 -3.34300048
## [24,]         NA        NA  1.0497982  0.77485532
## [25,] -3.2411387 -2.391492         NA          NA
## [26,]  1.7421789  1.485163         NA          NA
## [27,]         NA        NA -3.9158641 -4.14178218
## [28,]         NA        NA  2.9313577  1.51307707
## [29,]         NA        NA  0.3721062 -0.71655061
## [30,]         NA        NA  2.5544887  2.02524424
```

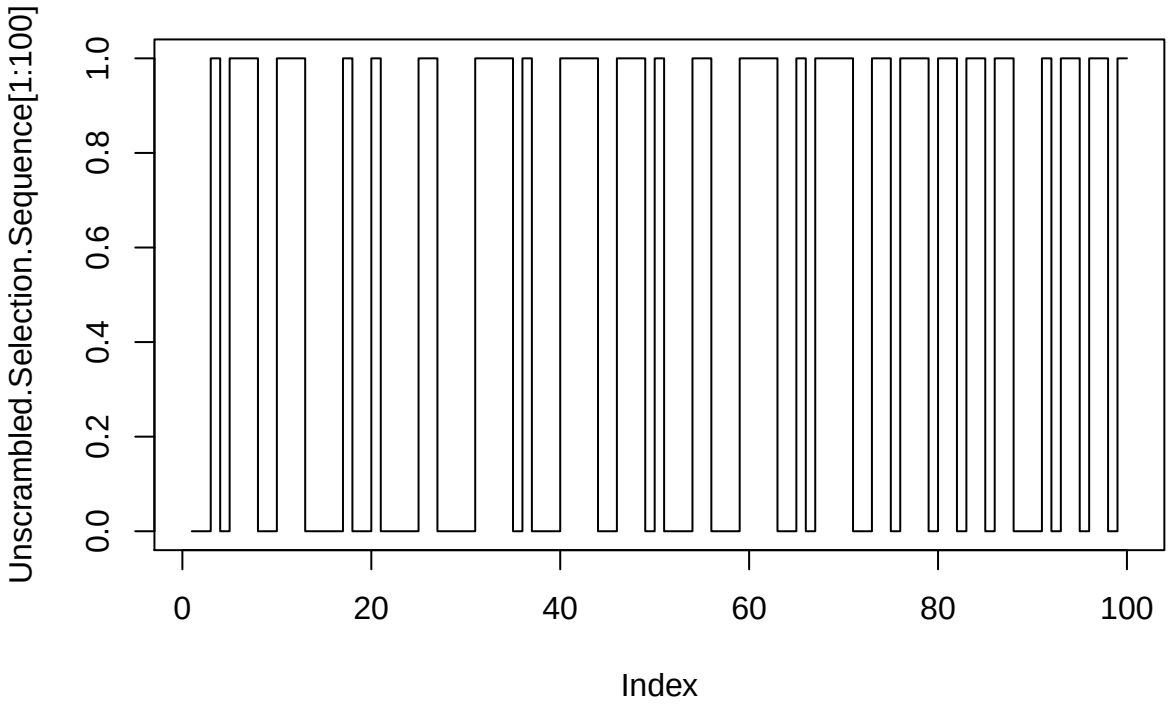
And plot the two clusters

```
matplot(LinearModelData$Input, cbind(LinearModel1.Recovered[, 2], LinearModel2.Recovered[,2]),
        type = "p", col = c("green", "blue"), pch = 19, ylab = "Separated Subsamples")
```




plot the unscrambled selection

```
plot(Unscrambled.Selection.Sequence[1:100], type = "s")
```




### 3.3 Confusion Matrix

There is a common measure for comparison of the estimated Unscrambled.Selection.Sequence and the true selection sequence that may be known from the training data set. The measure is called confusion matrix. Confusion matrix for comparison of Unscrambled.Selection.Sequence estimated in the project with the true selection sequence used to create the data is:

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

***don't run because they don't give you Selection.Sequence.true***

```
#cm<-confusionMatrix(Unscrambled.Selection.Sequence,Selection.Sequence.true)$table
#cm
```

The elements C(Pred,Act) of the table are:

True negative C(0,0) True positive C(1,1) False negative C(0,1) False positive C(1,0) Then there are several characteristics of prediction quality with following definitions:

Accuracy: P(Prediction correct) Sensitivity: P(Pred=1|Act=1) Specificity: P(Pred=0|Act=0) Balanced accuracy: 12(Sensitivity + Specificity) Calculate accuracy, sensitivity, specificity and balanced accuracy for the confusion table above.

```
accuracy<-(450+458)/1000
sensitivity<-(458/500)
```

```
specificity<-450/500
balancedAccuracy<-.5*(specificity+sensitivity)
c(Accuracy=accuracy,
  Sensitivity=sensitivity,
  Specificity=specificity,
  Balanced=balancedAccuracy)
```

```
##    Accuracy Sensitivity Specificity    Balanced
##       0.908       0.916       0.900       0.908
```

Think the coder got sensitivity, specificity mixed up in their coding

# 4 Estimating models for subsamples

## 4.1 Fitting models

Now estimate the linear models from the subsamples.

```
lm1x<-LinearModel1.Recovered[,1]
lm1y<-LinearModel1.Recovered[,2]
lm1<-data.frame(x=lm1x,y=lm1y)
LinearModel1.Recovered.lm<-lm(y~x, data = lm1, na.action = na.omit)
lm2x<-LinearModel2.Recovered[,1]
lm2y<-LinearModel2.Recovered[,2]
lm2<-data.frame(x=lm2x,y=lm2y)
LinearModel2.Recovered.lm<-lm(y~x, data = lm2, na.action = na.omit)
#LinearModel1.Recovered.lm<-lm(LinearModel1.Recovered[,2]~LinearModel1.Recovered[,1], data=LinearModel1
#LinearModel2.Recovered.lm<-lm(LinearModel2.Recovered[,2], LinearModel1.Recovered[,1], data=LinearModel
```

## 4.2 Comparison of the models

Compare the results of fitting of the first recovered linear model:

```
summary(LinearModel1.Recovered.lm)$coefficients
```

```
##               Estimate  Std. Error  t value      Pr(>|t|)
## (Intercept) 0.7331544 0.019479048 37.63810 8.716159e-149
## x           0.8012446 0.008346118 96.00207  0.000000e+00
```

```
summary(LinearModel1.Recovered.lm)$sigma
```

```
## [1] 0.4389739
```

```
summary(LinearModel1.Recovered.lm)$df
```

```
## [1]   2 506   2
```

```
summary(LinearModel1.Recovered.lm)$r.squared
```

```
## [1] 0.9479552
```

```
summary(LinearModel1.Recovered.lm)$adj.r.squared
```

```
## [1] 0.9478524
```

and the second recovered linear model:

```
summary(LinearModel2.Recovered.lm)$coefficients
```

```
##               Estimate  Std. Error   t value      Pr(>|t|)
## (Intercept) -0.6941222 0.020008001 -34.69223 4.708656e-134
## x            0.8107406 0.008586561  94.41971 1.557398e-316
```

```
summary(LinearModel2.Recovered.lm)$sigma
```

```
## [1] 0.4433244
```

```
summary(LinearModel2.Recovered.lm)$df
```

```
## [1]   2 490   2
```

```
summary(LinearModel2.Recovered.lm)$r.squared
```

```
## [1] 0.9479005
```

```
summary(LinearModel2.Recovered.lm)$adj.r.squared
```

```
## [1] 0.9477942
```

with the summary of the fit to the whole sample.

The sigma parameters:

```
c(summary(Estimated.LinearModel)$sigma,
  summary(LinearModel1.Recovered.lm)$sigma,
  summary(LinearModel2.Recovered.lm)$sigma)
```

```
## [1] 0.8388930 0.4389739 0.4433244
```

The R^2:

```
c(summary(Estimated.LinearModel)$r.squared,
  summary(LinearModel1.Recovered.lm)$r.squared,
  summary(LinearModel2.Recovered.lm)$r.squared)
```

```
## [1] 0.8307611 0.9479552 0.9479005
```

The F-statistics:

```
rbind(LinearModel=summary(Estimated.LinearModel)$fstatistic,
      LinearModel1.Recovered=summary(LinearModel1.Recovered.lm)$fstatistic,
      LinearModel2.Recovered=summary(LinearModel2.Recovered.lm)$fstatistic)
```

```
##                            value numdf dendf
## LinearModel             4898.989     1   998
## LinearModel1.Recovered 9216.397     1   506
## LinearModel2.Recovered 8915.082     1   490
```

Here is how we can calculate p-values of F-test using cumulative probability function of F-distribution:

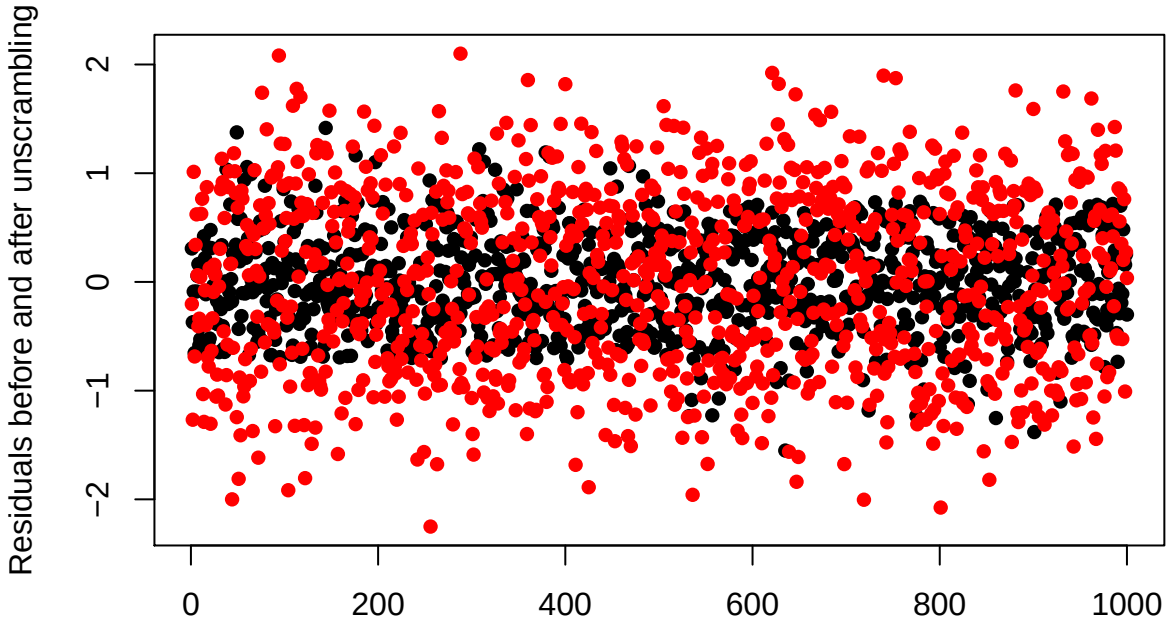
```
c(LinearModel=pf(summary(Estimated.LinearModel)$fstatistic[1],
                 summary(Estimated.LinearModel)$fstatistic[2],
                 summary(Estimated.LinearModel)$fstatistic[3],lower.tail = FALSE),
  LinearModel1.Recovered=pf(summary(LinearModel1.Recovered.lm)$fstatistic[1],
                            summary(LinearModel1.Recovered.lm)$fstatistic[2],
                            summary(LinearModel1.Recovered.lm)$fstatistic[3],lower.tail = FALSE),
  LinearModel2.Recovered=pf(summary(LinearModel2.Recovered.lm)$fstatistic[1],
                            summary(LinearModel2.Recovered.lm)$fstatistic[2],
                            summary(LinearModel2.Recovered.lm)$fstatistic[3],lower.tail = FALSE))
```

```
##             LinearModel.value LinearModel1.Recovered.value
##                  0.000000e+00                  0.000000e+00
## LinearModel2.Recovered.value
##                 1.557398e-316
```

The numbers may not look exactly the same as in summary() because of the precision limitation.

Compare the combined residuals of the two separated models with the residuals of Estimated.LinearModel

```
matplot(cbind(MixedModel.residuals=c(summary(LinearModel1.Recovered.lm)$residuals,
                                     summary(LinearModel2.Recovered.lm)$residuals),
              Single.Model.residuals=summary(Estimated.LinearModel)$residuals),
        type="p",pch=16,ylab="Residuals before and after unscrambling")
```




and estimate the different standard deviations:

```
apply(cbind(MixedModel.residuals=c(summary(LinearModel1.Recovered.lm)$residuals,
                                   summary(LinearModel2.Recovered.lm)$residuals),
            Single.Model.residuals=summary(Estimated.LinearModel)$residuals),2,sd)
```

```
##   MixedModel.residuals Single.Model.residuals
##              0.4404568              0.8384730
```

***What is the difference between the quality of fit?*** The mixed models have a much lower standard deviation

***What is the difference between the two estimated models?***

***Try to guess how the model data were simulated and with what parameters?***