Week 4: Homework Assignment Analysis of Residuals of a Linear Model

John Navarro

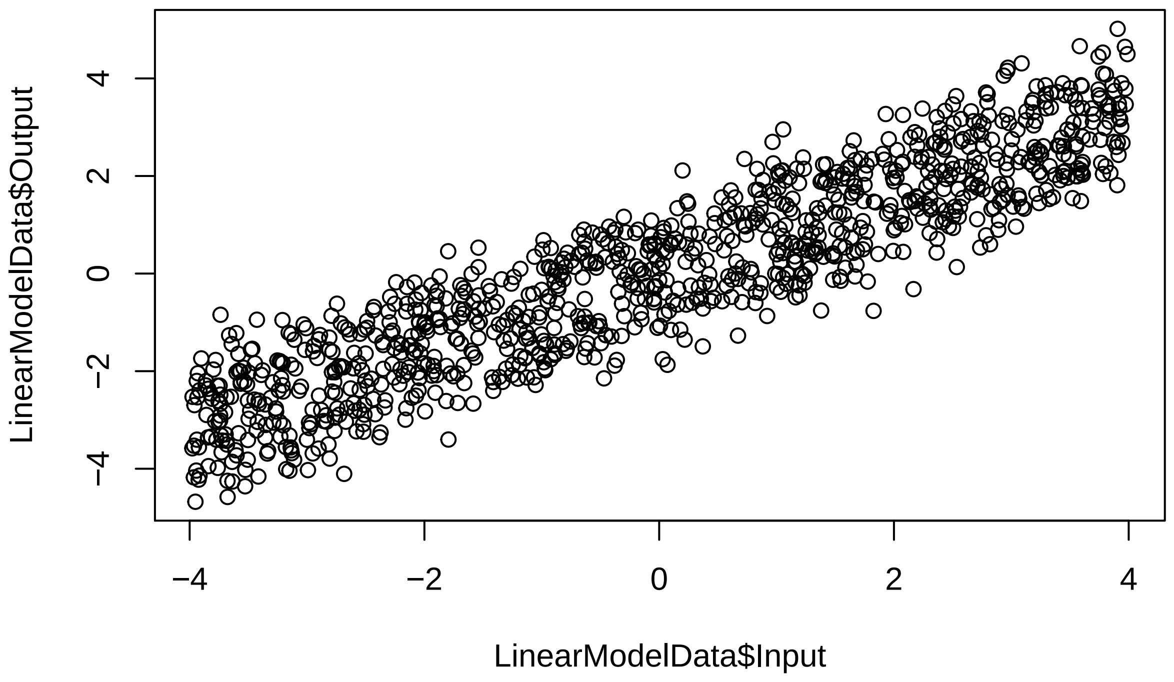
October 22, 2016

1 Data

```
##read in data and assign to LinearModelData, print head of df
datapath<-"C:/Users/JohntheGreat/Documents/MSCA/StatisticalAnalysis/Week4/Assignments"
LinearModelData<-read.csv(file=paste(datapath,"ResidualAnalysisProjectData_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
```

```
##plot the data
plot(LinearModelData$Input,LinearModelData$Output)
```



2 Fitting linear model

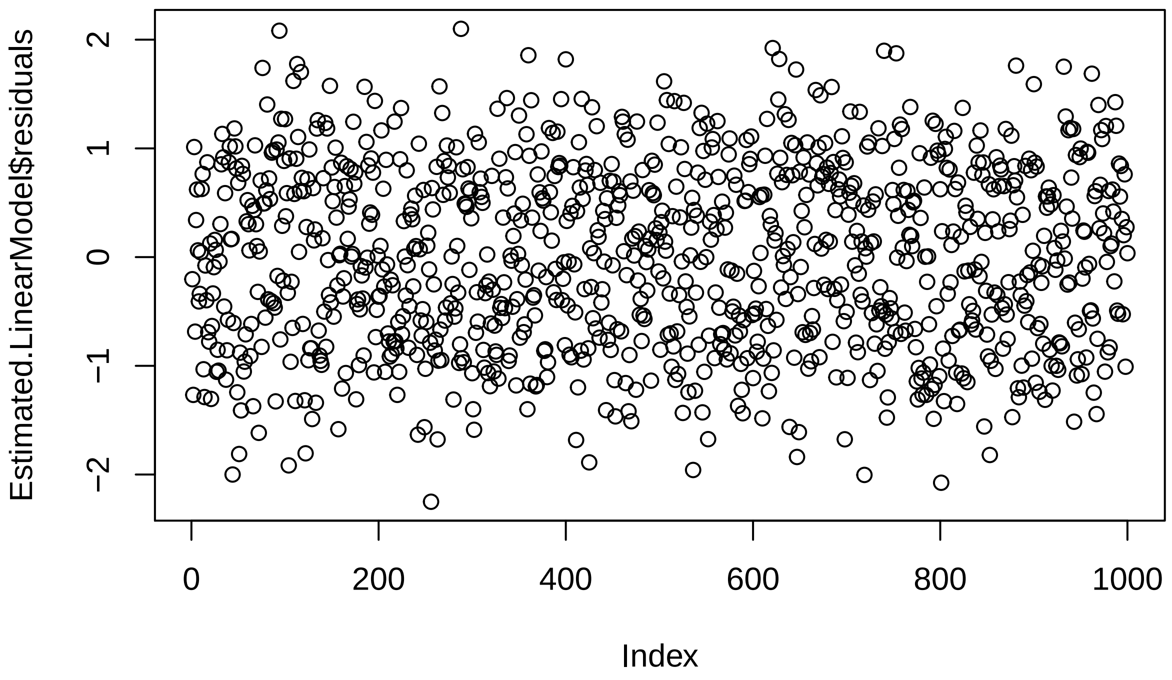
```
Estimated.LinearModel <- lm(Output ~ Input,data=LinearModelData)
names(Estimated.LinearModel)
```

```
##  [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? The residuals are the difference between the model's Y output (Estimated.LinearModelSfitted.values) and data's Y output(LinearModelDataSOutput) *3. Find out what are fitted.values* The fitted.values are the specific outputs of the model, given X

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



```
##Estimated.LinearModel$fitted.values
```

2.2 Object of 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 By looking at the Coefficients in the summary, we can see that the probability of the t value of the intercept is insignificant. However, the slope is statistically significant and we can reject the null hypothesis that the slope equals zero and accept that there is a linear relationship between Input and Output. This is confirmed by the levels of R-squared.

```
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? This calls the standard error of the Residuals from the Summary output.

```
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)

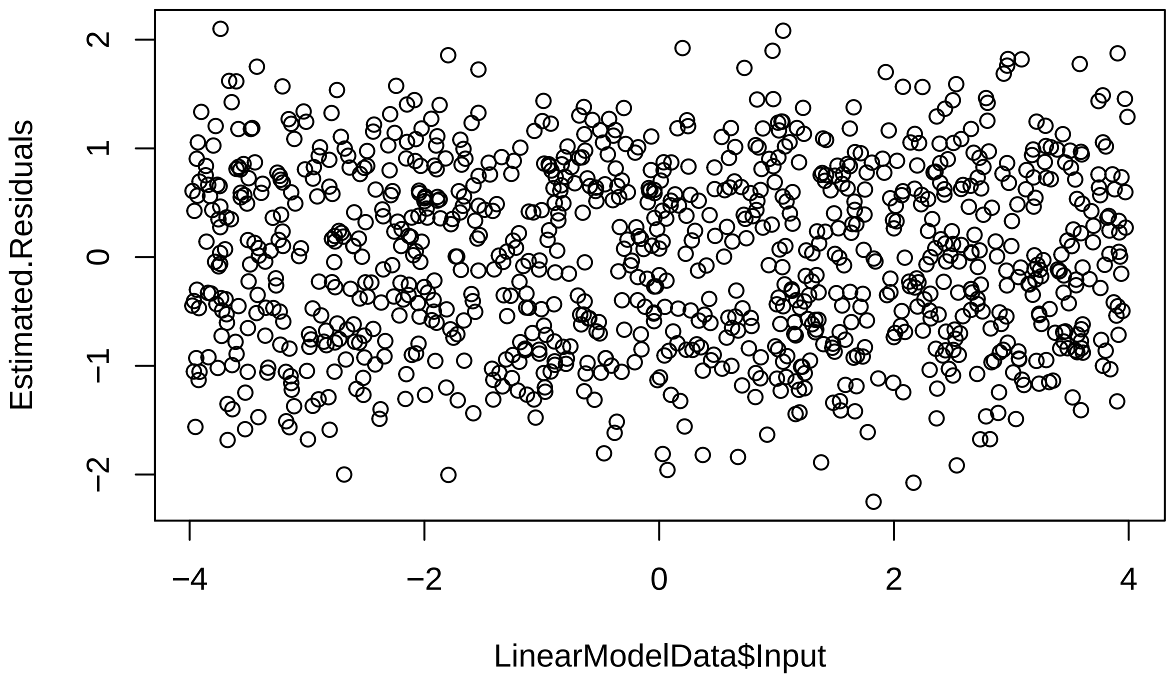
```
sigmaSquared.byVar <- (999/998) *var(Estimated.LinearModel$residuals)
sigmaSquared.bySum <-  sum((Estimated.LinearModel$residuals)^2)/998
c(sigmaSquared.byVar=sigmaSquared.byVar,sigmaSquared.bySum=sigmaSquared.bySum,
  fromModel=summary(Estimated.LinearModel)$sigma^2)
```

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

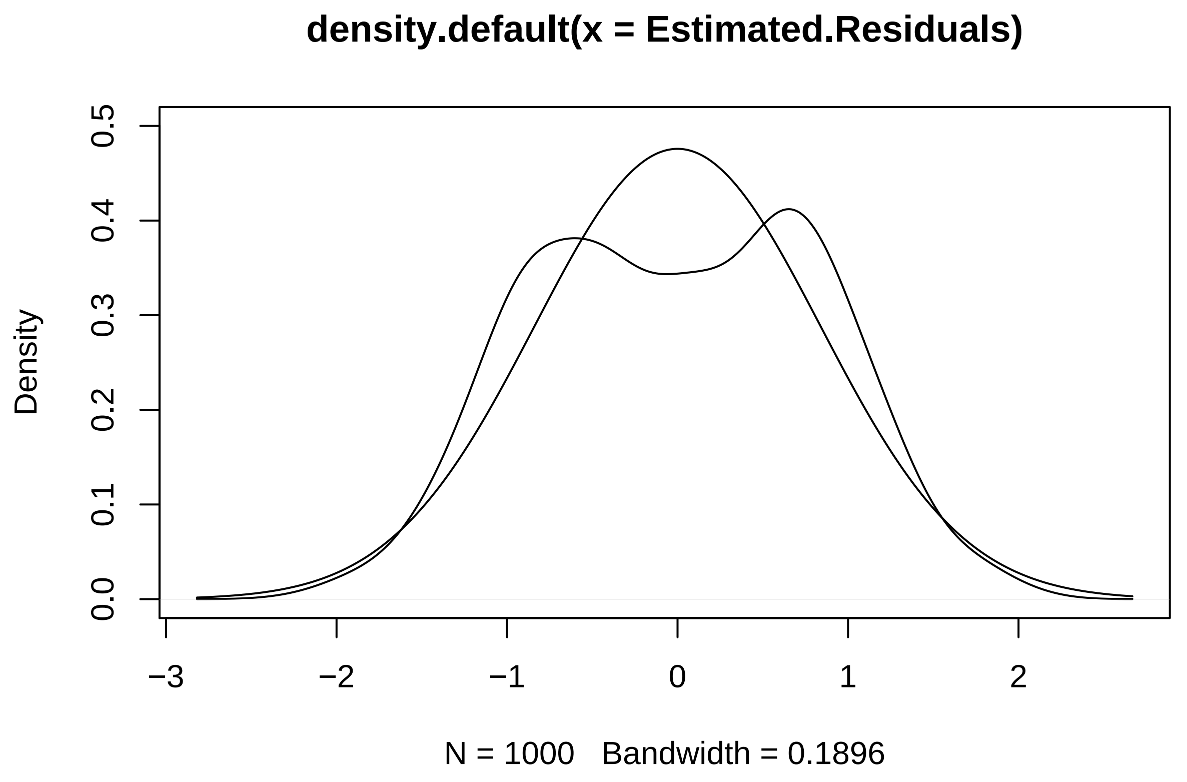
3 Analysis of residuals

3.1 Residuals of the model

```
##Observe the residuals, plot them against the input.
Estimated.Residuals <- Estimated.LinearModel$residuals
plot(LinearModelData$Input, Estimated.Residuals)
```



```
## assign the density of Estimated.Residuals, Plot the density function and overlay with a
##normal distributio using the mean and sd from Estimated.Residuals
Probability.Density.Residuals <- density(Estimated.Residuals)
plot(Probability.Density.Residuals, ylim = c(0, 0.5))
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? On a plot, the residuals appear to be a mix of uniform and normal distribution. When you plot the density function, you can see that there are 2 clusters of residuals on each side of the mean.

3.2 Clustering the 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
```

```
##Create Sequence of Residuals
Unscrambled.Selection.Sequence <- c()
for (i in 1:1000) {
  if(Estimated.Residuals[i]<0){
    Unscrambled.Selection.Sequence[i] <- 0
  } else{Unscrambled.Selection.Sequence[i] <- 1
  }
}
head(Unscrambled.Selection.Sequence,30)
```

```
##  [1] 0 0 1 0 1 1 1 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 0 0 1 1 0 0 0 0
```

```
LinearModelData.Seq <- cbind(LinearModelData,Unscrambled.Selection.Sequence)
##Create matrices with original data if applicable, otherwise fill element with NA
LinearModel1.Recovered <- LinearModelData
for (i in 1:1000){
  if(LinearModelData.Seq$Unscrambled.Selection.Sequence[i] ==0) {
    LinearModel1.Recovered$Input[i] <- NA
    LinearModel1.Recovered$Output[i] <- NA
  } else {
    LinearModel1.Recovered$Input[i] <- LinearModelData$Input[i]
    LinearModel1.Recovered$Output[i] <- LinearModelData$Output[i]
  }
}
head(LinearModel1.Recovered)
```

```
##       Input   Output
## 1        NA       NA
## 2        NA       NA
## 3 0.6475581 1.559734
## 4        NA       NA
## 5 1.9921334 1.958417
## 6 1.7534556 2.049381
```

```
LinearModel2.Recovered <- LinearModelData
for (i in 1:1000){
  if(LinearModelData.Seq$Unscrambled.Selection.Sequence[i] ==1) {
    LinearModel2.Recovered$Input[i] <- NA
    LinearModel2.Recovered$Output[i] <- NA
  } else {
    LinearModel2.Recovered$Input[i] <- LinearModelData$Input[i]
    LinearModel2.Recovered$Output[i] <- LinearModelData$Output[i]
  }
}
head(LinearModel2.Recovered)
```

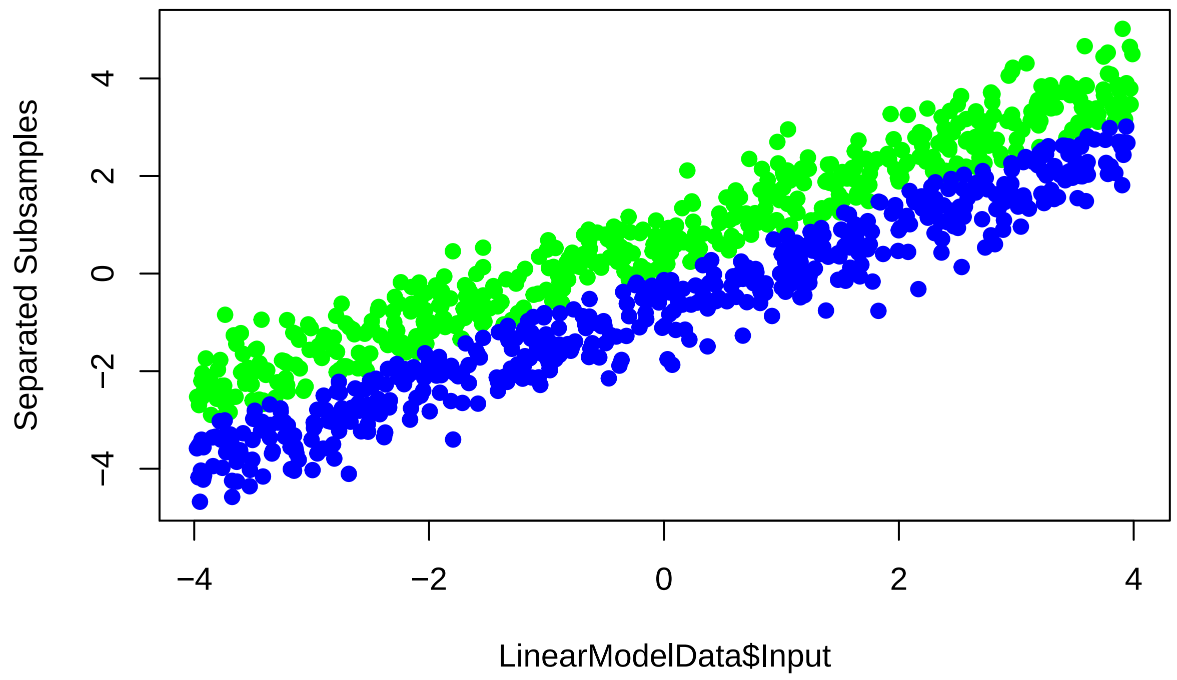
```
##       Input    Output
## 1  3.666433  2.747905
## 2 -2.519442 -3.242035
## 3        NA        NA
## 4  2.443962  1.292082
## 5        NA        NA
## 6        NA        NA
```

```
##combine into one matrix
head(cbind(LinearModel1.Recovered,LinearModel2.Recovered),30)
```

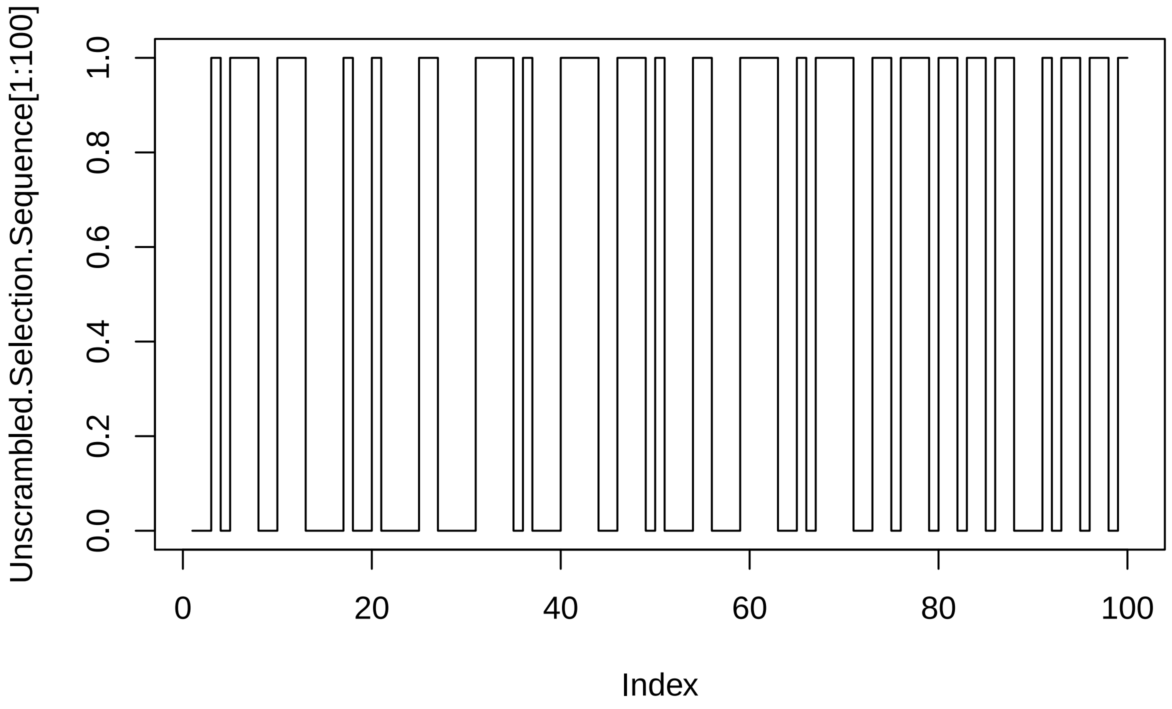
```
##         Input   Output      Input      Output
## 1          NA       NA  3.6664327  2.74790517
## 2          NA       NA -2.5194424 -3.24203530
## 3   0.6475581 1.559734         NA          NA
## 4          NA       NA  2.4439621  1.29208230
## 5   1.9921334 1.958417         NA          NA
## 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
```

```
##Plot two clusters

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



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



3.3 Confusion matrix

```
suppressWarnings(library(caret))
```

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

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

```
##cm<-confusionMatrix(Unscrambled.Selection.Sequence,Selection.Sequence.true)$table
##cm
##create confusion matrix by hand, calculate the characteristics of
##prediction quality
cm.hand <- matrix(c(450,50,42,458),nrow=2, ncol=2)
dimnames(cm.hand) <- list(c("Pred=0","Pred=1"),c("Ref=0","Ref=1"))
cm.hand
```

```
##        Ref=0 Ref=1
## Pred=0   450    42
## Pred=1    50   458
```

```r
accuracy <- (cm.hand[1,1]+cm.hand[2,2])/1000
sensitivity <-cm.hand[1,1]/500
specificity <- cm.hand[2,2]/500
balancedAccuracy <- 0.5 *(sensitivity + specificity)
c(Accuracy=accuracy, Sensitivity=sensitivity, Specificity=specificity, Balanced=balancedAccuracy)
```

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

4 Estimating models for subsamples

4.1 Fitting models

```r
#Now estimate the linear models from the subsamples.
LinearModel1.Recovered.lm <- lm(LinearModel1.Recovered$Output ~ LinearModel1.Recovered$Input)
LinearModel2.Recovered.lm <- lm(LinearModel2.Recovered$Output ~ LinearModel2.Recovered$Input)
```

4.2 Comparison of the models

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

```
##                               Estimate  Std. Error  t value      Pr(>|t|)
## (Intercept)                  0.7331544 0.019479048 37.63810 8.716159e-149
## LinearModel1.Recovered$Input 0.8012446 0.008346118 96.00207  0.000000e+00
```

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

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

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

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

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

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

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

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

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

```
##                                 Estimate  Std. Error   t value
## (Intercept)                   -0.6941222 0.020008001 -34.69223
## LinearModel2.Recovered$Input  0.8107406 0.008586561  94.41971
##                                   Pr(>|t|)
## (Intercept)                   4.708656e-134
## LinearModel2.Recovered$Input 1.557398e-316
```

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

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

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

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

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

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

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

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

```r
##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
```

```r
##The Rho Squared:
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
```

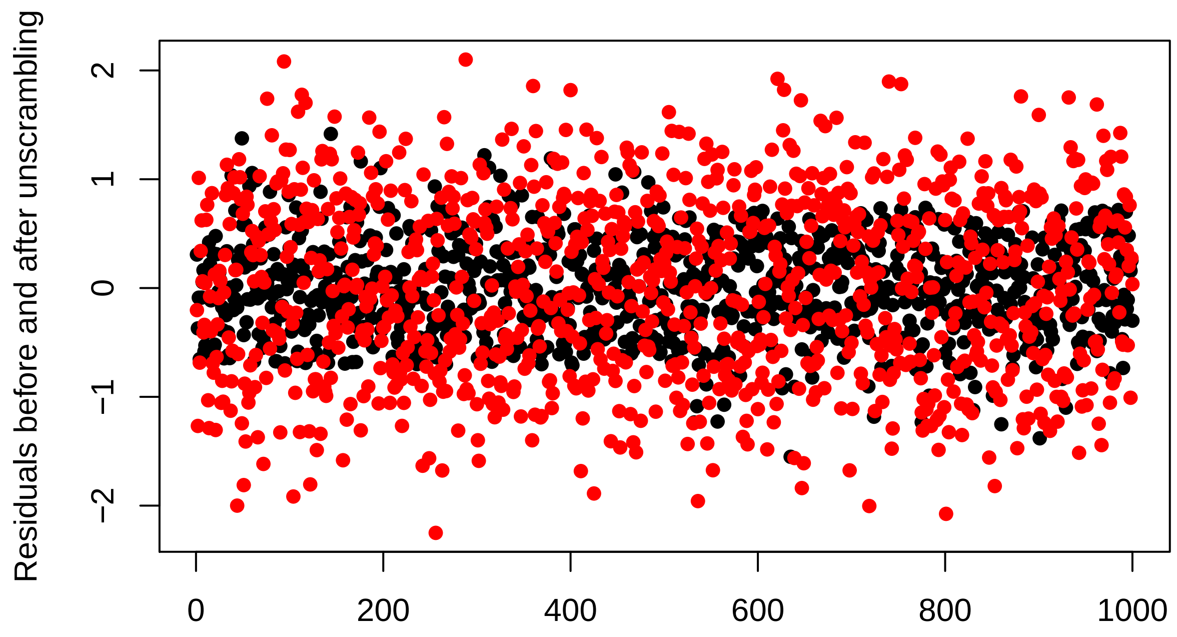
```r
##The F-statistics
rbind(LinearModel=summary(Estimated.LinearModel)$fstatistic,
      LinearModel1.Recovered=summary(LinearModel1.Recovered.lm)$fstatistic,
      LinearModel2.Recovered=summary(LinearModel2.Recovered.lm)$fstatistic)
```

```
##                           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-distributi
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
```

```
##Compare the combined residuals of the two separated models with the residuals of Estimated.LinearMode
## Plot residuals
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")
```



```
## Estimate 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 Model gives us a much smaller residual standard error(~0.44) than the original Single Model(0.839). The Mixed Model also has much higher correlations (0.947) than the Single Model(0.831). It appears that separating the residuals gives us a much better fit than using one linear model. *What is the difference between the two estimated models?* The difference is how we separated the residuals. This leads to higher correlations and lower sigmas for the Mixed Model approach. *Try to guess how the model data were simulated and with what parameters?* I believe that the residuals were calculated using random distribution, but with two different sets of means.