# Week3_Assignment

*John Navarro*

*October 14, 2016*

```
##Set Slope and Intercept and nSample
a<-.8; b<-.1
nSample<-1000
```

## 1. Model 1

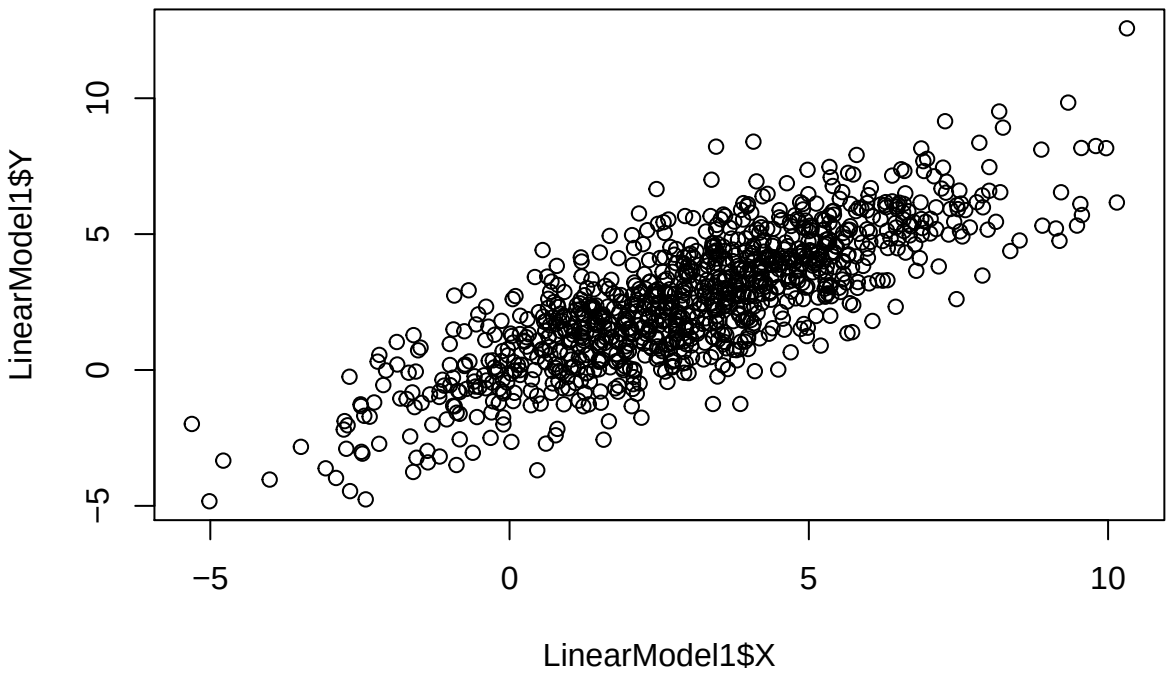
```
##simulate Eps by setting seed and using rnorm()
set.seed(1112131415)
Eps <- rnorm(nSample, mean = 0, sd = 1.5)

##Simulate X by setting seed and using rnorm()
set.seed(111)
X1 <- rnorm(nSample, mean = 3, sd = 2.5)
## Create Y1 as a linear function
Y1 <- a*X1 + b + Eps
##use all X1 and Y1 to create a dataframe called LinearModel1
LinearModel1 <- as.data.frame(cbind(Y = Y1, X = X1, Eps = Eps))
##Take standard deviations of X1 sample and Eps sample
sd.X <- sd(X1)
sd.Eps <- sd(Eps)
```

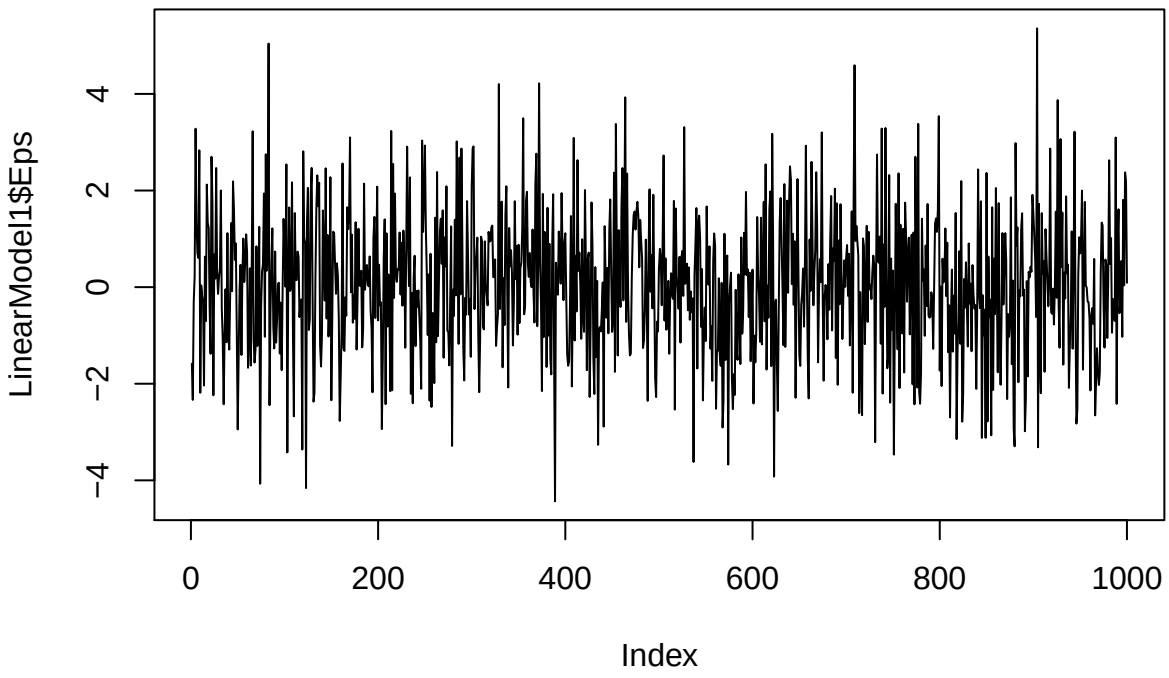
```
head(LinearModel1)
```

```
##            Y         X        Eps
## 1  1.3856455  3.588052 -1.5847959
## 2 -0.4957909  2.173160 -2.3343191
## 3  1.5640592  2.220940 -0.3126932
## 4 -1.8813610 -2.755864  0.2233303
## 5  5.4377138  2.572810  3.2794659
## 6  4.1192926  3.350696  1.3387362
```

```
##Plot X vs Y
plot(LinearModel1$X,LinearModel1$Y)
```




```
##Plot the residuals of the model
plot(LinearModel1$Eps,type="l")
```

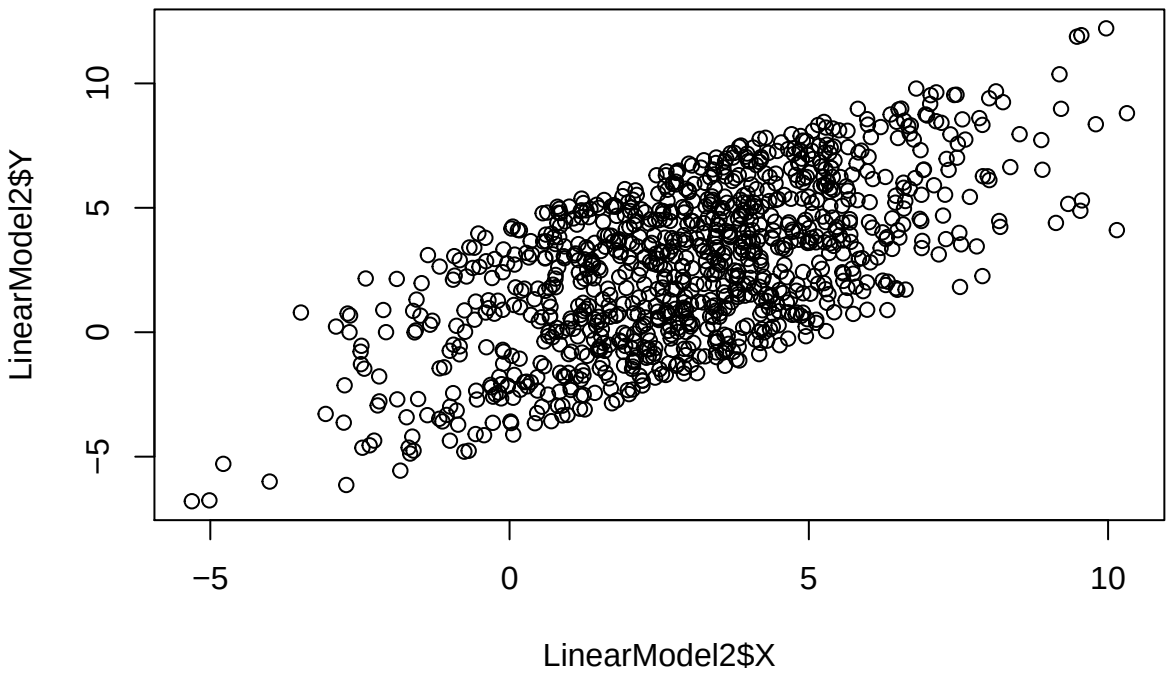



#2. Model 2

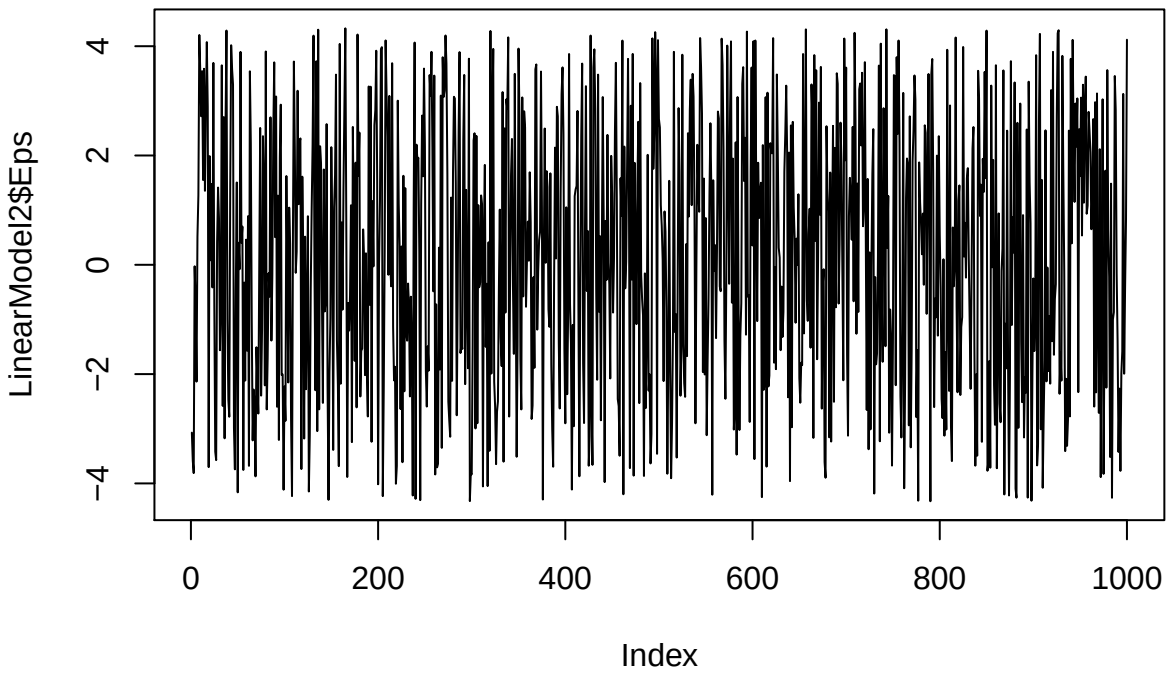
```
#simulate Eps
set.seed(1112131415)
Eps <- runif(nSample, min = -4.33, max = 4.33)
##Use the same realization of X as Model 1
Y1 <- a*X1 + b + Eps
LinearModel2 <- as.data.frame(cbind(Y = Y1, X = X1, Eps = Eps))
head(LinearModel2)
```

```
##            Y         X         Eps
## 1 -0.1007155  3.588052 -3.07115695
## 2 -1.7495480  2.173160 -3.58807627
## 3 -1.9351307  2.220940 -3.81188300
## 4 -2.1321719 -2.755864 -0.02748057
## 5  1.4432271  2.572810 -0.71502082
## 6  0.6424869  3.350696 -2.13806956
```

```
## Plot X vs Y and plot Eps
plot(LinearModel2$X,LinearModel2$Y)
```




```
plot(LinearModel2$Eps,type="l")
```

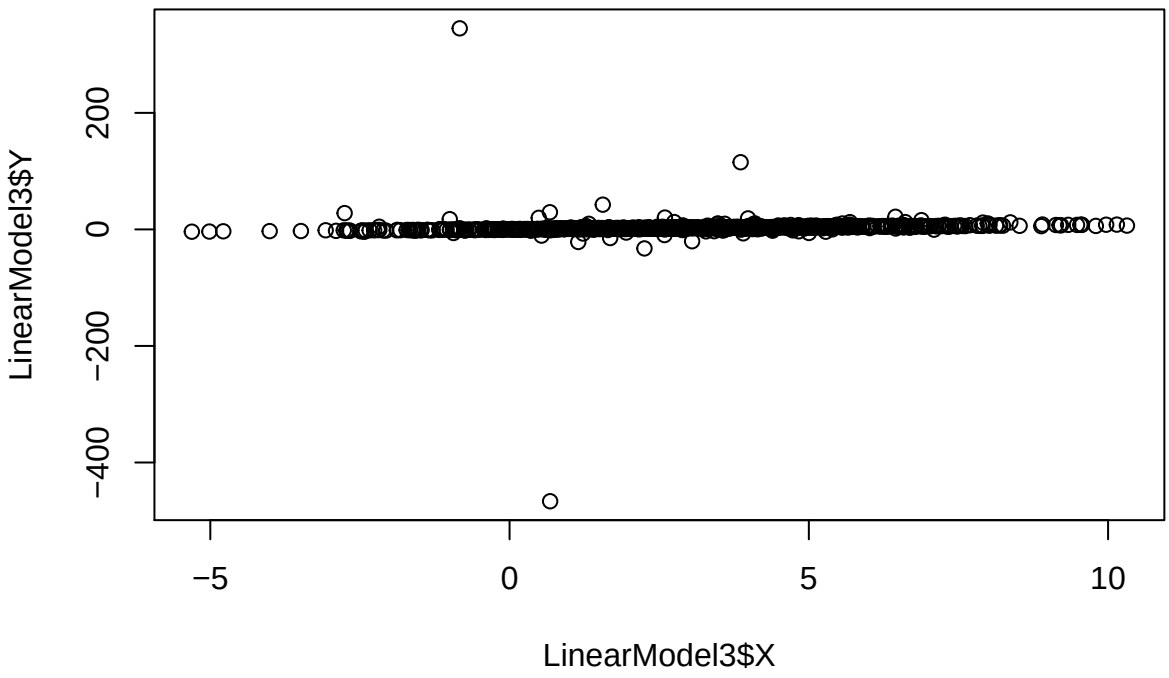



#3. Model 3

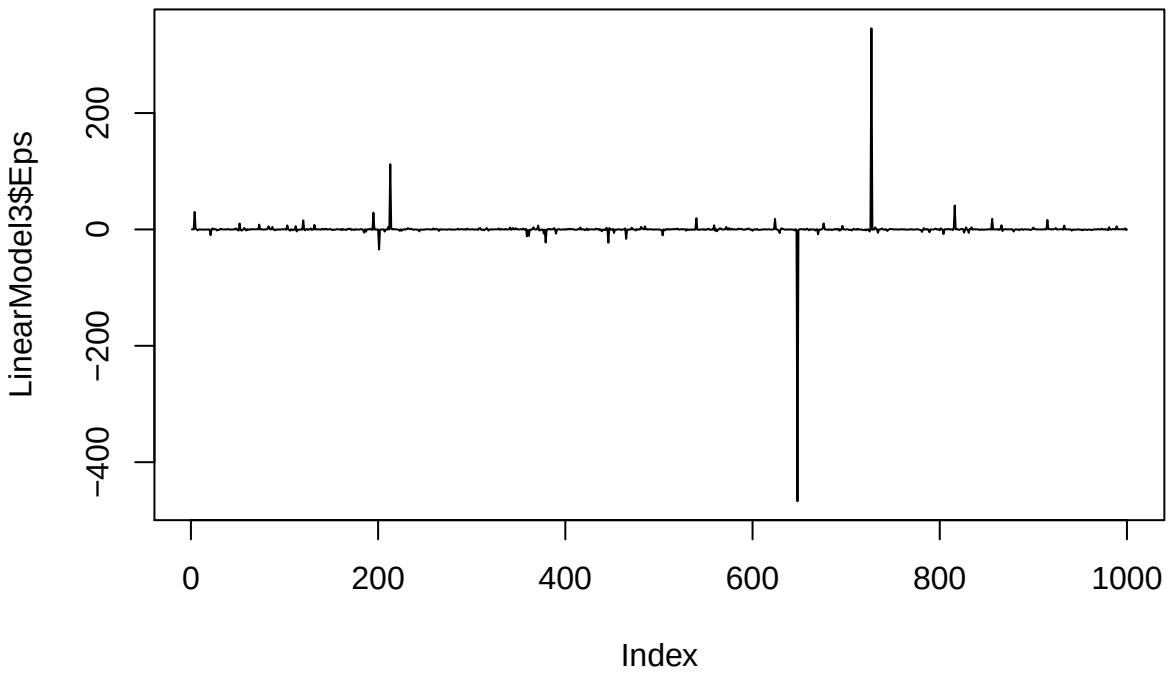
```
#simulate Eps
set.seed(1112131415)
Eps <- rcauchy(nSample, location = 0, scale = 0.3)
##Use the same realization of X as Model 1
Y1 <- a*X1 + b + Eps
LinearModel3 <- as.data.frame(cbind(Y = Y1, X = X1, Eps = Eps))
head(LinearModel3)
```

```
##           Y         X         Eps
## 1  3.117834  3.588052  0.14739288
## 2  1.921281  2.173160  0.08275234
## 3  1.933813  2.220940  0.05706081
## 4 27.987178 -2.755864 30.09186936
## 5  3.288758  2.572810  1.13051049
## 6  3.086476  3.350696  0.30591976
```

```
## Plot linearmodel 3 and residuals
plot(LinearModel3$X,LinearModel3$Y)
```




```
plot(LinearModel3$Eps,type="l")
```




```
##Calculate the standard deviation of the residuals
sd(LinearModel3$Eps)
```

```
## [1] 18.98625
```

```
##Generate 5 more Eps samples and their standard deviations
Eps1<-rcauchy(n=nSample,location=0,scale=.3)
Eps2<-rcauchy(n=nSample,location=0,scale=.3)
Eps3<-rcauchy(n=nSample,location=0,scale=.3)
Eps4<-rcauchy(n=nSample,location=0,scale=.3)
Eps5<-rcauchy(n=nSample,location=0,scale=.3)
c(sd(Eps1),sd(Eps2),sd(Eps3),sd(Eps4),sd(Eps5))
```

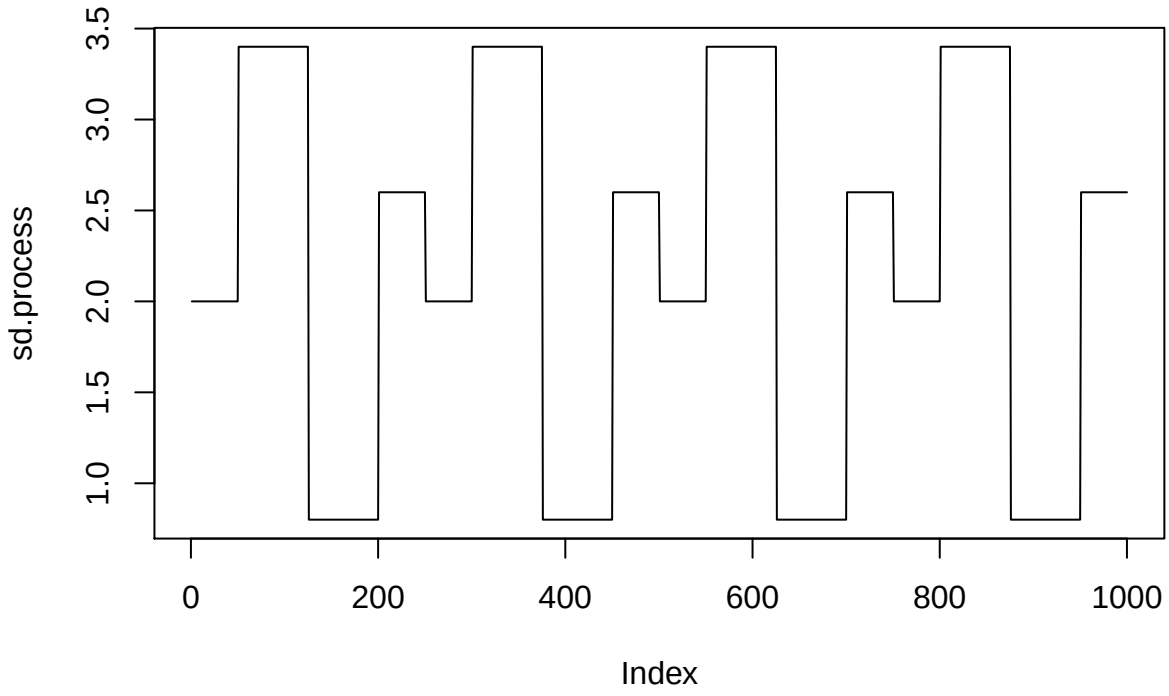
```
## [1]   4.357834   7.316044 279.660160   4.778542   6.561620
```

*How do you interpret this observation?* Cauchy distributions are very random and extreme, it is not suprising that different samples will give vastly differnt standard deviations. #4. Model 4 Create the process of standard deviations in which the first 50 observations have sigma=2, followed by 75 observations with sigma=3.4, followed by 75 observations with sigma=0.8 and concluded by 50 observations with sigma=2.6.

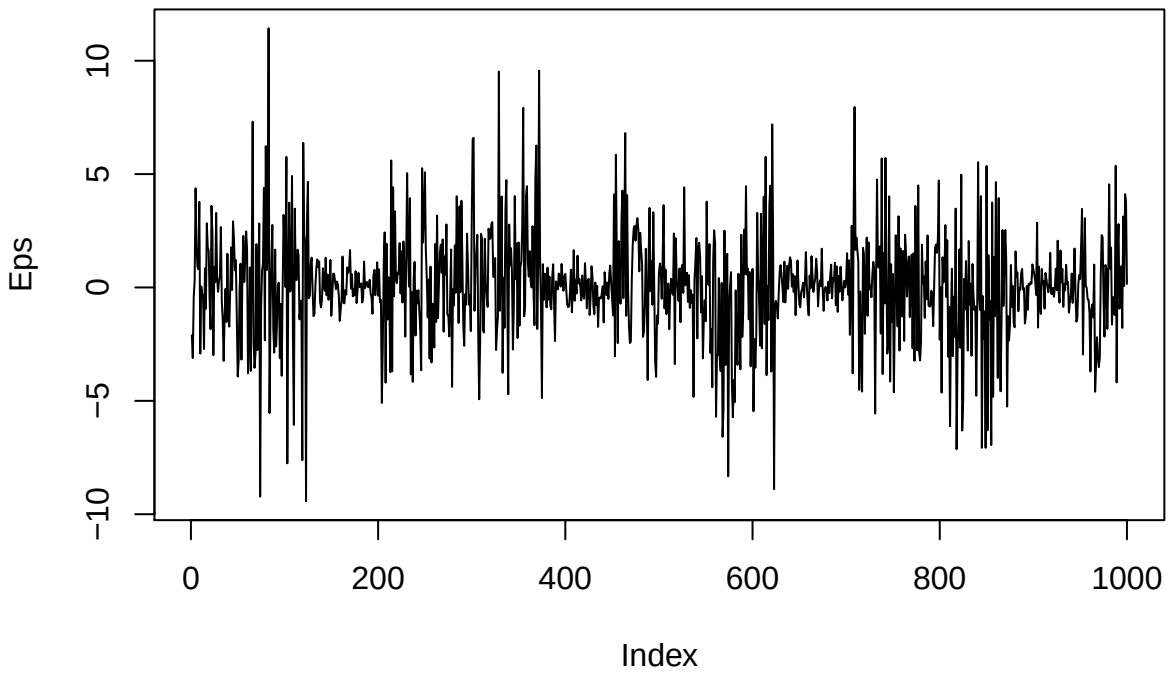
```
sd.Values<-c(2,3.4,.8,2.6)
sd.process<-rep(c(rep(sd.Values[1],50),
                  rep(sd.Values[2],75),
                  rep(sd.Values[3],75),
```

```
                  rep(sd.Values[4],50)),
            4)

plot(sd.process,type="l")
```

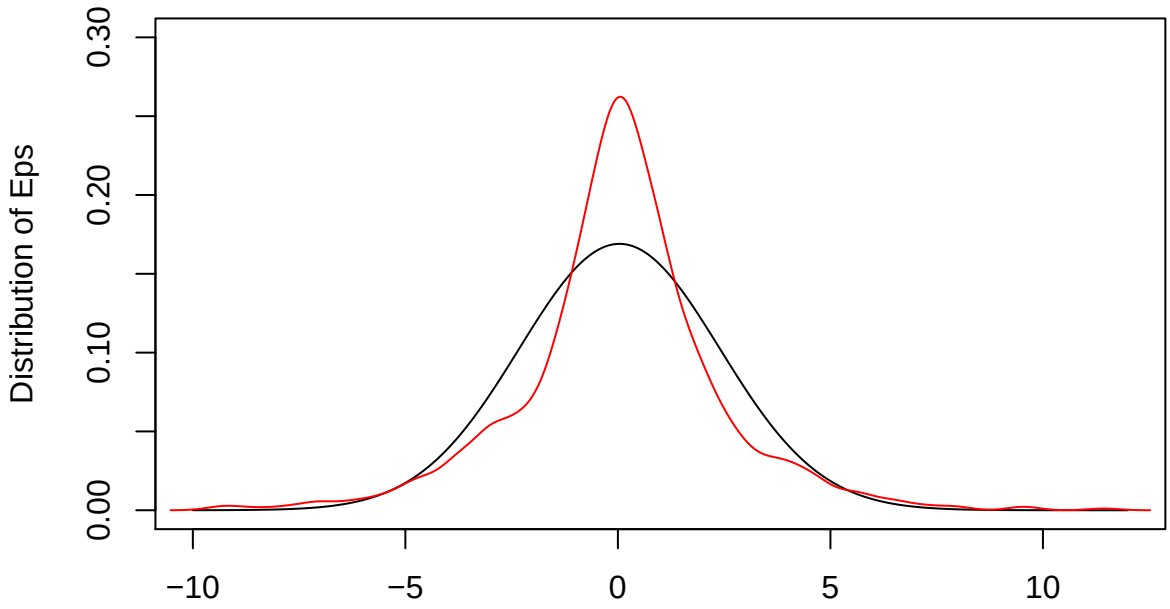



```
##Create linear model residuals  with the changing SDs and plot
set.seed(1112131415);
Eps<-rnorm(nSample)*sd.process
plot(Eps,type="l")
```

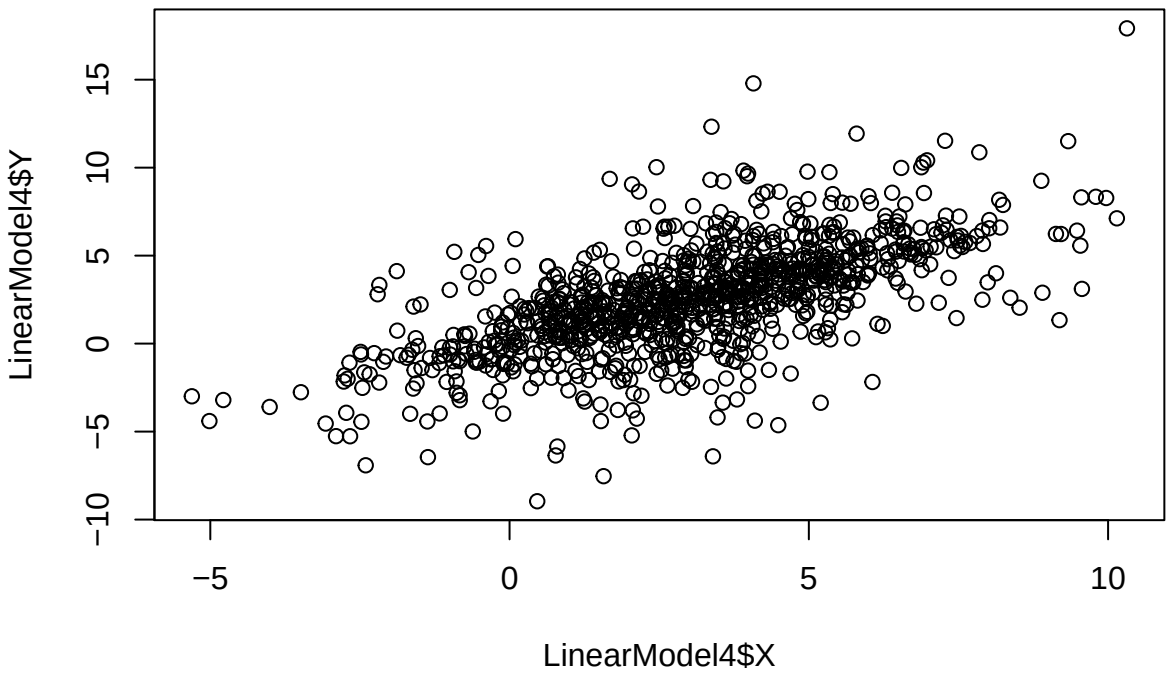



Observe how heteroscedasticity transforms normal distribution into leptokurtic distribution

```
Xvariable<-(100*floor(min(Eps))):(100*ceiling(max(Eps)))
Xvariable<-Xvariable/100
# Plot the sample distribution and the theo. distribution
plot(Xvariable,dnorm(Xvariable,mean=mean(Eps),sd=sd(Eps)),type="l",
      ylim=c(0,.3),col="black",ylab="Distribution of Eps",xlab="")
lines(density(Eps),col="red")
```




```
##Generate LinearModel4 and plot
Y1<-a*X1+b+Eps
LinearModel4<-as.data.frame(cbind(Y=Y1,X=X1))
plot(LinearModel4$X,LinearModel4$Y)
```




#5. Effect of Residual Distribution on Correlation

```
##calculate theoretical
Theoretical.Rho.Squared<-(a*sd.X)^2/((a*sd.X)^2+sd.Eps^2)
Theoretical.Rho.Squared
```

```
## [1] 0.6467077
```

```
##compare with the correlations from the 4 models
c(cor(LinearModel1$X,LinearModel1$Y)^2,
  cor(LinearModel2$X,LinearModel2$Y)^2,
  cor(LinearModel3$X,LinearModel3$Y)^2,
  cor(LinearModel4$X,LinearModel4$Y)^2)
```

```
## [1] 0.635937885 0.410346727 0.009230536 0.405022505
```

*How do you interpret the results?*

The correlation for Linear Model 1 is very close to the value for Theoretical.Rho.Squared. While the other 3 models produce correlations that are different and lower than the theoretical value. The correlation from the Cauchy model is vastly different and extremely low. While the correlations for the uniform and heteroscedastic models are lower than normal model correlation, they are still somewhat close. This tells me that these two models explain some of the correlation that is seen, but not all of it.

# 6. Estimation of Linear Model

```
##Create a linear regression using X&Y data from LinearModel1
m1<-lm(Y~X,data=LinearModel1)
summary(m1)
```

```
##
## Call:
## lm(formula = Y ~ X, data = LinearModel1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4709 -0.9800  0.0003  0.9537  5.3112
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.21129    0.07307   2.891  0.00392 **
## X            0.78109    0.01871  41.753  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.46 on 998 degrees of freedom
## Multiple R-squared:  0.6359, Adjusted R-squared:  0.6356
## F-statistic:  1743 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
names(summary(m1))
```

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

```
##look at the parameters
summary(m1)$r.squared
```

```
## [1] 0.6359379
```

```
summary(m1)$coeff
```

```
##              Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) 0.2112857 0.07307137  2.891497  3.917289e-03
## X           0.7810888 0.01870749 41.752730 3.364178e-221
```

```
summary(m1)$sigma^2
```

```
## [1] 2.132694
```

```
var(summary(m1)$residuals)
```

```
## [1] 2.130559
```

```
##reconcile sigmaEstimate with m1$sigma by normalizing with respect to deg of freedom
var(summary(m1)$residuals)*999/998
```

```
## [1] 2.132694
```

Estimate the same parameters using the method of moments directly.

```
##Solve for aEstimate, then bEstimate, then sigmaEstimate using LinearModel1 data
aEstimate <- cov(LinearModel1$X,LinearModel1$Y)/var(LinearModel1$X)
bEstimate <- mean(LinearModel1$Y) - (aEstimate * mean(LinearModel1$X))
sigmaEstimate <- sqrt(var(LinearModel1$Y) - ((aEstimate^2) * (var(LinearModel1$X))))
```

The result of estimation by method of moments is:

```
c(aEstimate,bEstimate,sigmaEstimate)
```

```
## [1] 0.7810888 0.2112857 1.4596435
```

Reconcile sigmaEstimate with m1$sigma.

```
##normalize m1$sigma^2 with respect to their degrees of freedom
c(sigmaMetodMoments=sigmaEstimate,sigmaLinearModel=summary(m1)$sigma)
```

```
## sigmaMetodMoments  sigmaLinearModel
##          1.459643          1.460375
```

```
reconciled.sigmaLinearModel <- sqrt(((summary(m1)$sigma)^2) * (998/999))
c(sigmaMethodMoments = sigmaEstimate,reconciledSigmaLinModel = reconciled.sigmaLinearModel)
```

```
##      sigmaMethodMoments reconciledSigmaLinModel
##                1.459643               1.459643
```

# 7. Fit lm() to the the Rest of Linear Models

Compare the differences between the assumptions of the 4 models and tell how they change the model behavior and estimated parameters.

```
##fit the other models into linear regressions and look at parameters
m2<-lm(Y~X,data=LinearModel2)
summary(m2)
```

```
##
## Call:
## lm(formula = Y ~ X, data = LinearModel2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5927 -2.1379 -0.0049  2.1687  4.2386
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.16263    0.12308   1.321    0.187
## X            0.83042    0.03151  26.354   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.46 on 998 degrees of freedom
## Multiple R-squared:  0.4103, Adjusted R-squared:  0.4098
## F-statistic: 694.5 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
names(summary(m2))
```

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

```
summary(m2)$coeff
```

```
##              Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) 0.1626258 0.12307972  1.321305  1.867027e-01
## X           0.8304188 0.03151046 26.353749 1.326871e-116
```

```
summary(m2)$sigma
```

```
## [1] 2.459821
```

```
summary(m2)$r.squared
```

```
## [1] 0.4103467
```

```
summary(m2)$df
```

```
## [1]   2 998   2
```

```
m3<-lm(Y~X,data=LinearModel3)
summary(m3)
```

```
##
## Call:
## lm(formula = Y ~ X, data = LinearModel3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -467.36   -0.43   -0.09    0.24  345.29
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   0.3629     0.9504   0.382  0.70270
## X             0.7420     0.2433   3.049  0.00235 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19 on 998 degrees of freedom
## Multiple R-squared:  0.009231,   Adjusted R-squared:  0.008238
## F-statistic: 9.298 on 1 and 998 DF,  p-value: 0.002355
```

```
names(summary(m3))
```

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

```
summary(m3)$coeff
```

```
##              Estimate Std. Error   t value    Pr(>|t|)
## (Intercept) 0.3628702  0.9504455 0.3817896 0.702698690
## X           0.7419727  0.2433299 3.0492458 0.002354668
```

```
summary(m3)$sigma
```

```
## [1] 18.99522
```

```
summary(m3)$r.squared
```

```
## [1] 0.009230536
```

```
summary(m3)$df
```

```
## [1]   2 998   2
```

```
m4<-lm(Y~X,data=LinearModel4)
summary(m4)
```

```
##
## Call:
## lm(formula = Y ~ X, data = LinearModel4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4956 -1.0531  0.0211  1.0844 11.4033
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.17245    0.11817   1.459    0.145
## X            0.78857    0.03025  26.065   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.362 on 998 degrees of freedom
## Multiple R-squared:  0.405,  Adjusted R-squared:  0.4044
## F-statistic: 679.4 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
names(summary(m4))
```

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

```
summary(m4)$coeff
```

```
##              Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) 0.1724487 0.11817209  1.459302  1.447967e-01
## X           0.7885655 0.03025403 26.064811 1.184840e-114
```

```
summary(m4)$sigma
```

```
## [1] 2.361739
```

```
summary(m4)$r.squared
```

```
## [1] 0.4050225
```

```
summary(m4)$df
```

```
## [1]   2 998   2
```

## Compare the differences between the assumptions of the 4 models and tell how they change the model behavior and estimated parameters.

The four linear models all use different distribution patterns to simulate the values of Eps in the equation Y=aX + B + Eps. The model 1 uses a normal distribution which gives us the closest match to the a, B, and sigma values attained from the data using Method of Moments. This tells us that the distribution of error terms in the data is normal.

The second linear model uses a uniform distribution to simulate the values of Eps between +/-4.33. This gives us slightly different parameters. The slope is slightly steeper, with a lower intercept, but a higher sigma. A uniform distribution should have a higher standard deviation for its error terms than a normal distribution, which would have them clustered around the mean, with small tails.

The third linear model uses a cauchy distribution to simulate the values of Eps. As we can see in the above plots, the model generates extreme outliers. This gives us a smaller slope, larger intercept, but a much, much higher sigma. Which tells us just how large the variance of the error terms are that were created with this model.

The fourth linear model uses a heteroscedastic distribution for its Eps values. This means that the standard deviation parameters vary during the generation of values by the normal distribution. Given this, we see that the Eps values have a larger standard deviation than a normal distribution. As well as a steeper slope and a lower intercept than the normal linear model.