Assignment 3

Matthew Dunne

April 10, 2018

```
a<-.8
b<-.1
nSample<-1000
```

1. Model 1

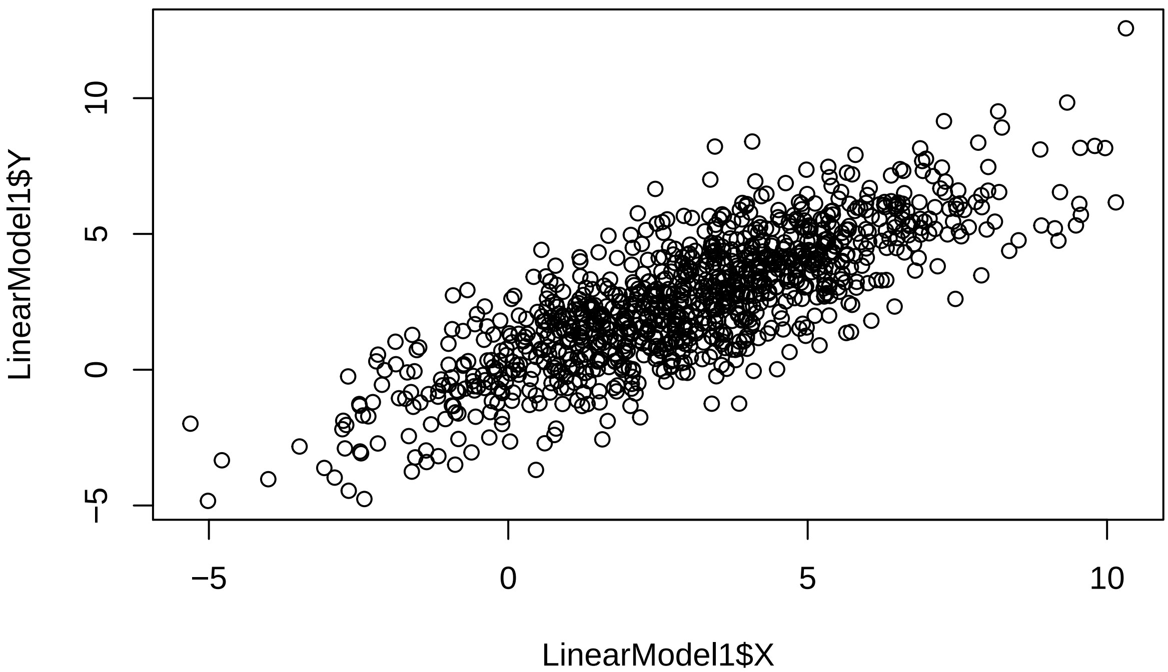
Simulate and plot Model1: input variable X ~ Norm(??=3,??=2.5); model residuals Eps ~ Norm(??=0,??=1.5)

```
set.seed(111)
X<-rnorm(nSample, 3, 2.5)
set.seed(1112131415)
Eps<-rnorm(nSample, 0, 1.5)
Y<-a*X+b+Eps
LinearModel1<-data.frame(Y=Y, X=X, Eps=Eps)
head(LinearModel1)
```

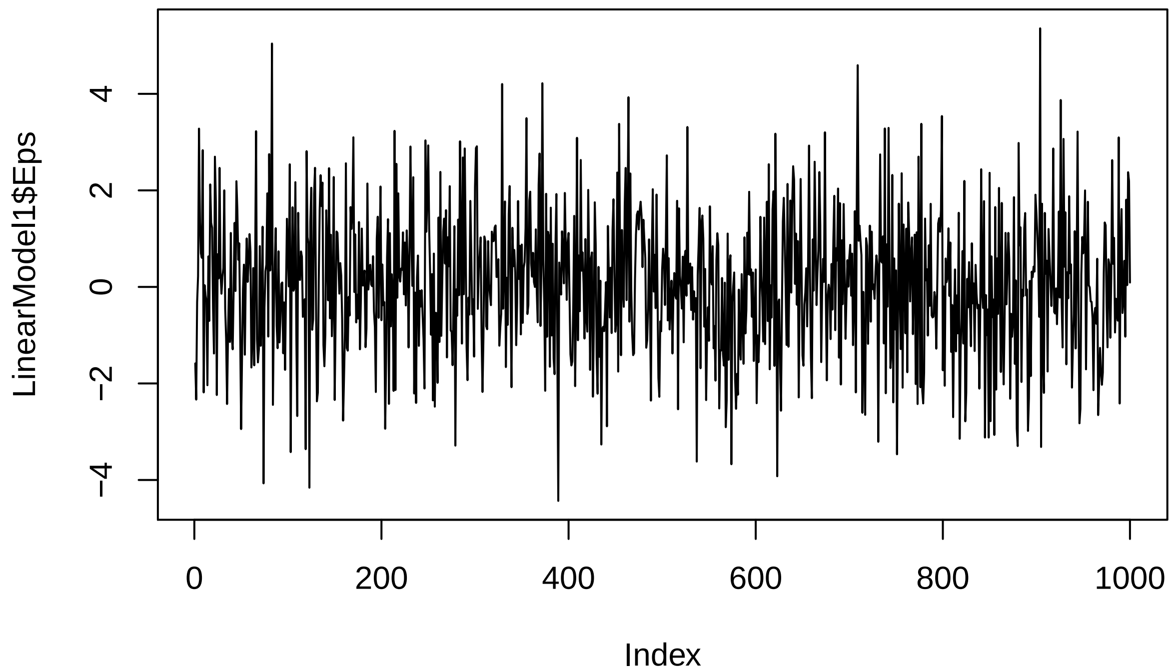
```
##            Y         X        Eps
## 1  1.3856455  3.588052 -1.5847959
## 2 -0.4957909  2.173160 -2.3343191
## 3  1.5640592  2.220940 -0.3126932
## 4 -1.8813610 -2.755864  0.2233303
## 5  5.4377138  2.572810  3.2794659
## 6  4.1192926  3.350696  1.3387362
```

Plot the model and the residuals of the model.

```
plot(LinearModel1$X, LinearModel1$Y)
```




```
plot(LinearModel1$Eps, type="l")
```



2. Model 2

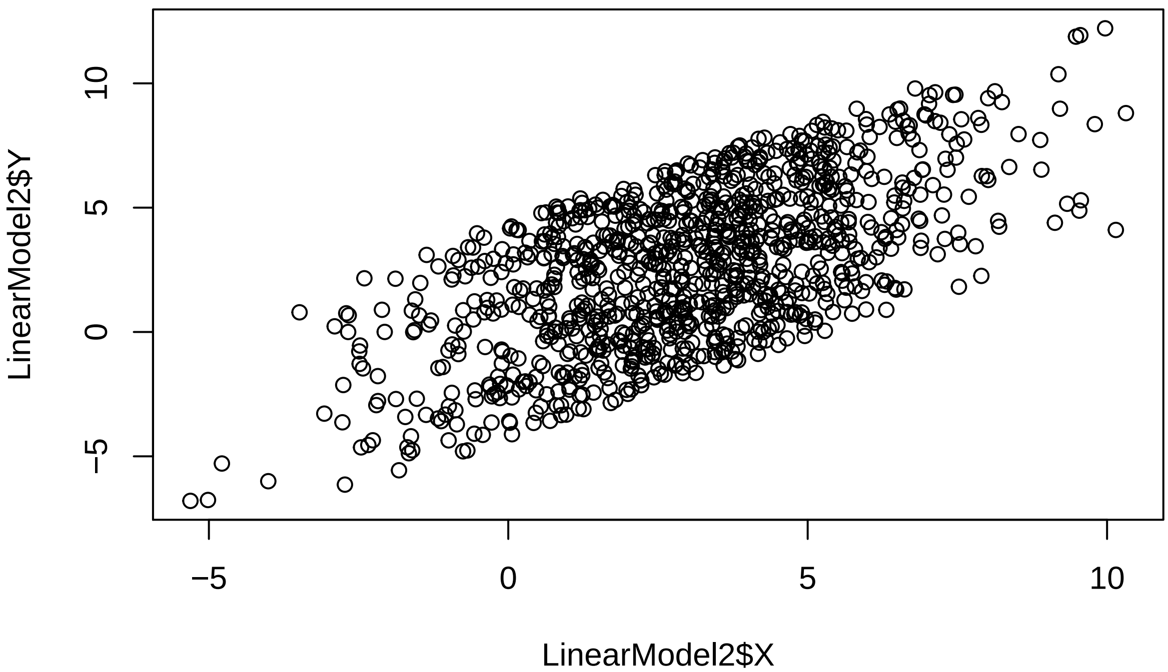
Simulate and plot Model2: input variable X ~ Norm(??=3,??=2.5); model residuals Eps ~ Unif(min=???4.33,max=4.33). Use the same realization of X as in the first model.

```
set.seed(111)
X<-rnorm(nSample, 3, 2.5)
set.seed(1112131415)
Eps<-runif(n=nSample, min = -4.33, max=4.33)
Y<-a*X+b+Eps
LinearModel2<-data.frame(Y=Y, X=X, Eps=Eps)
head(LinearModel2)
```

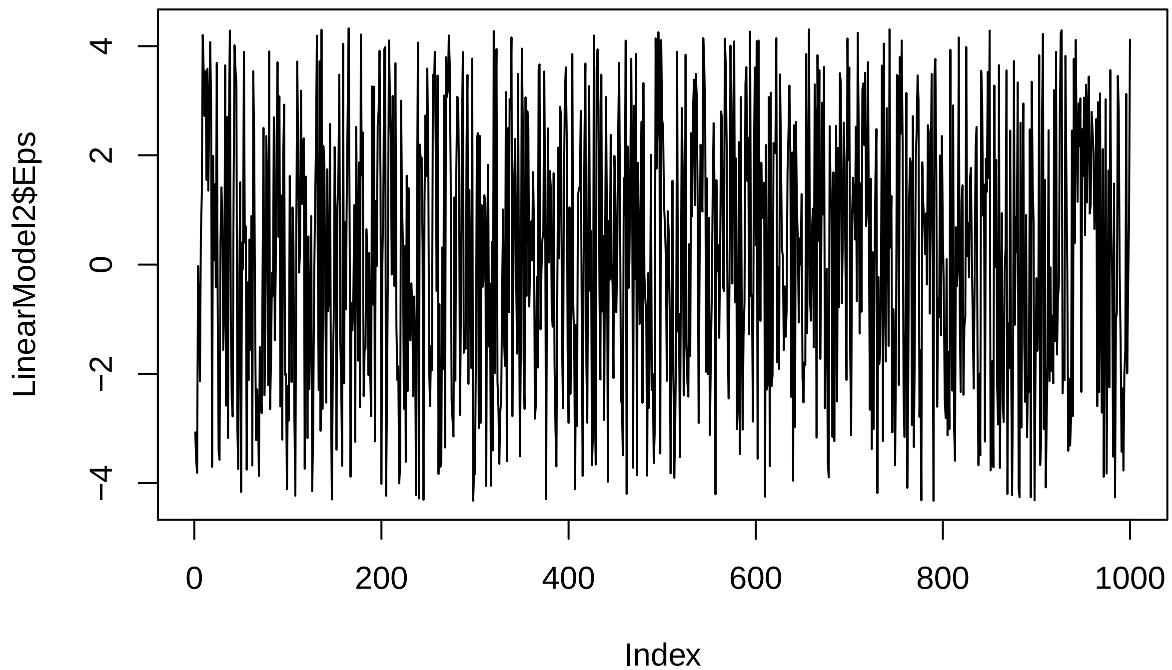
```
##             Y         X         Eps
## 1 -0.1007155  3.588052 -3.07115695
## 2 -1.7495480  2.173160 -3.58807627
## 3 -1.9351307  2.220940 -3.81188300
## 4 -2.1321719 -2.755864 -0.02748057
## 5  1.4432271  2.572810 -0.71502082
## 6  0.6424869  3.350696 -2.13806956
```

Plot the model and the residual.

```
plot(LinearModel2$X, LinearModel2$Y)
```




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



3. Model 3

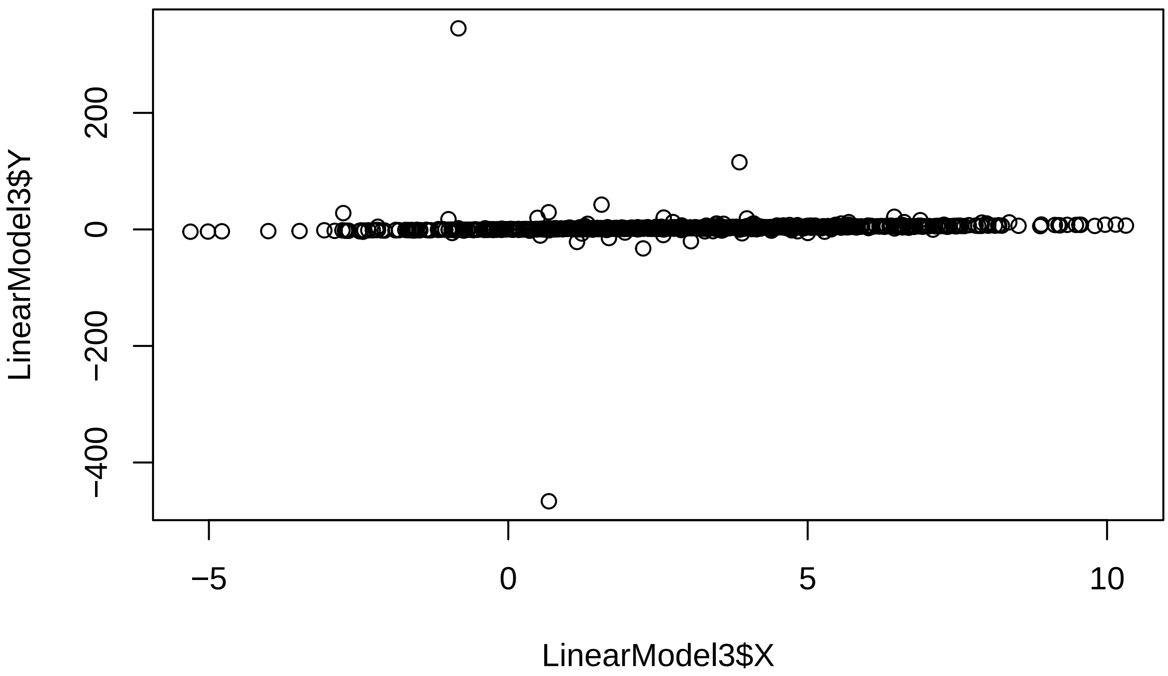
Simulate and plot Model3: input variable X ~ Norm^(??=3,??=2.5); model residuals Eps ~ Cauchi*(location=0,scale=0.3). Use the same realization of X as in the first model.

```
set.seed(111)
X<-rnorm(nSample, 3, 2.5)
set.seed(1112131415)
Eps<-rcauchy(n=nSample, location = 0, scale=0.3)
Y<-a*X+b+Eps
LinearModel3<-data.frame(X=X, Y=Y, Eps=Eps)
head(LinearModel3)
```

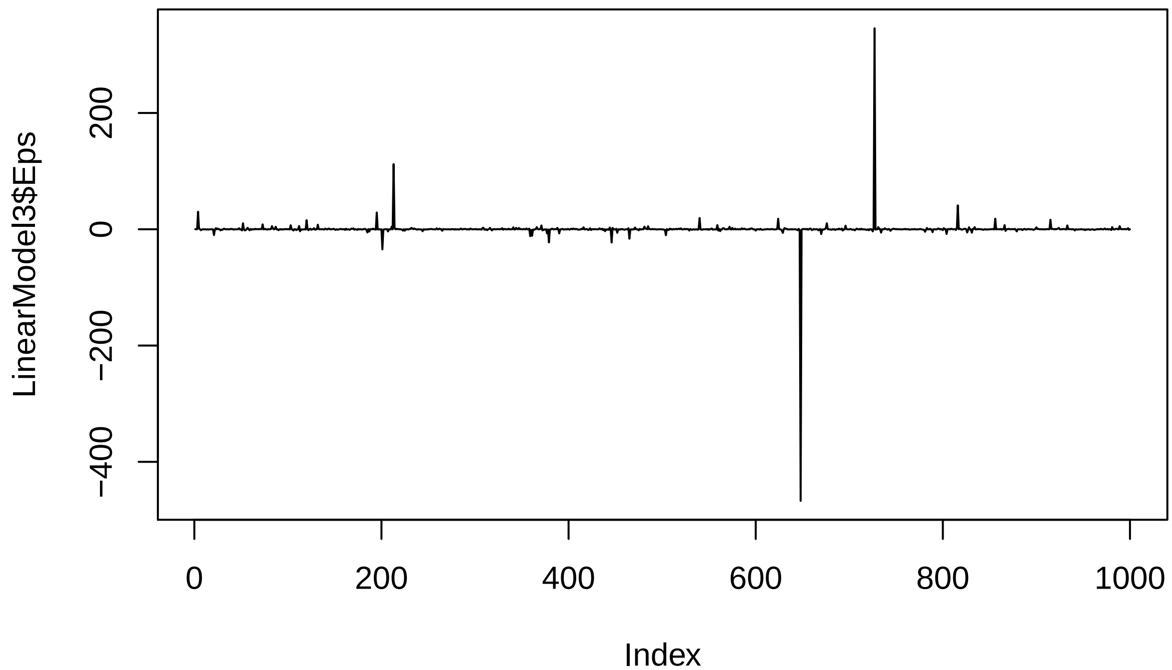
```
##           X         Y         Eps
## 1  3.588052  3.117834  0.14739288
## 2  2.173160  1.921281  0.08275234
## 3  2.220940  1.933813  0.05706081
## 4 -2.755864 27.987178 30.09186936
## 5  2.572810  3.288758  1.13051049
## 6  3.350696  3.086476  0.30591976
```

Plot the model and its residuals.

```
plot(LinearModel3$X, LinearModel3$Y)
```




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



Estimate the standard deviation of the residuals

```
sd(LinearModel3$Eps)
```

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

Generate another 5 samples of residuals without any seed specification and estimate standard deviations for each of them.

```
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
```

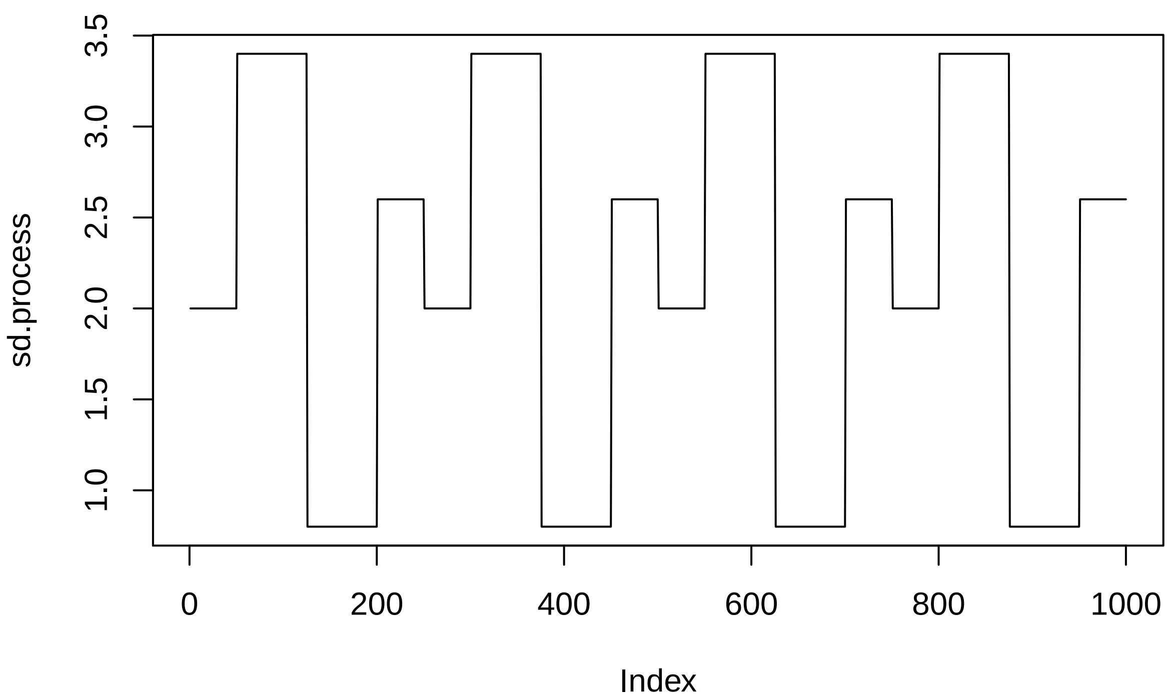
How do you interpret this observation?
Standard deviations of Cauchy distributions can vary greatly.

4. Model 4

Simulate and plot Model4: input variable X ~ Norm(??=3,??=2.5); model residuals Eps ~ a heteroscedastic process. Use the same realization of X as in the first model.

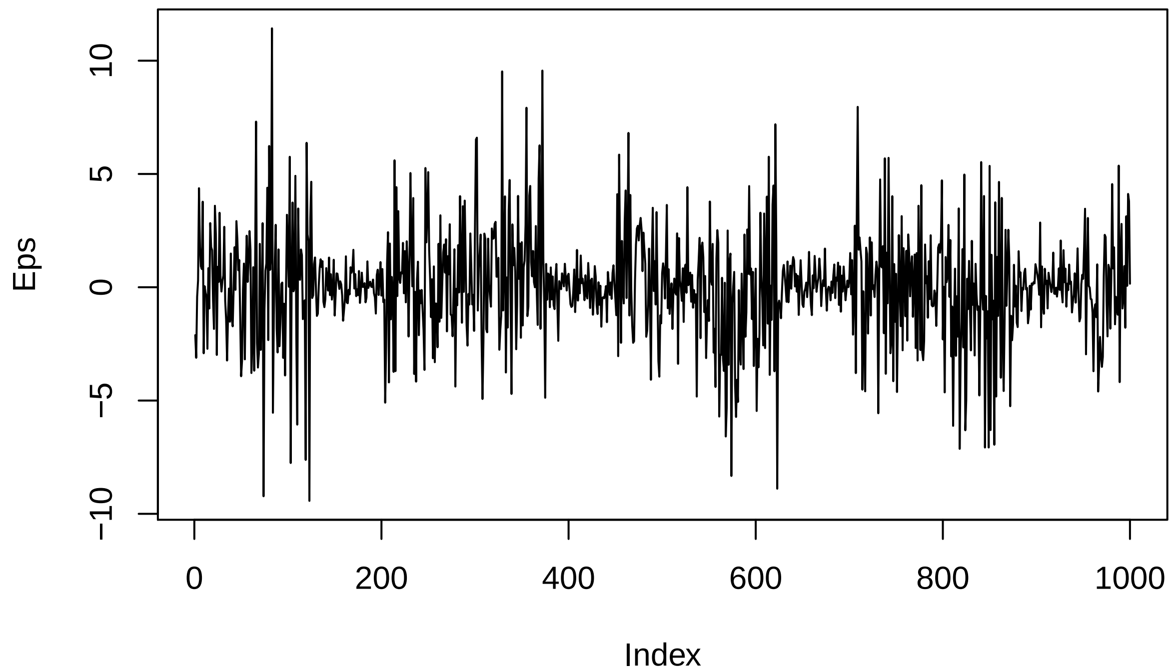
Create the process of standard deviations in which the first 50 observations have sigma=2, followed by 75 observations with sigma=3.4, followed by 75 observations with sigma=0.8 and concluded by 50 observations with sigma=2.6. Plot the trajectory of standard deviations of total length nSample=1000.

```
set.seed(111)
X<-rnorm(nSample, 3, 2.5)
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")
```



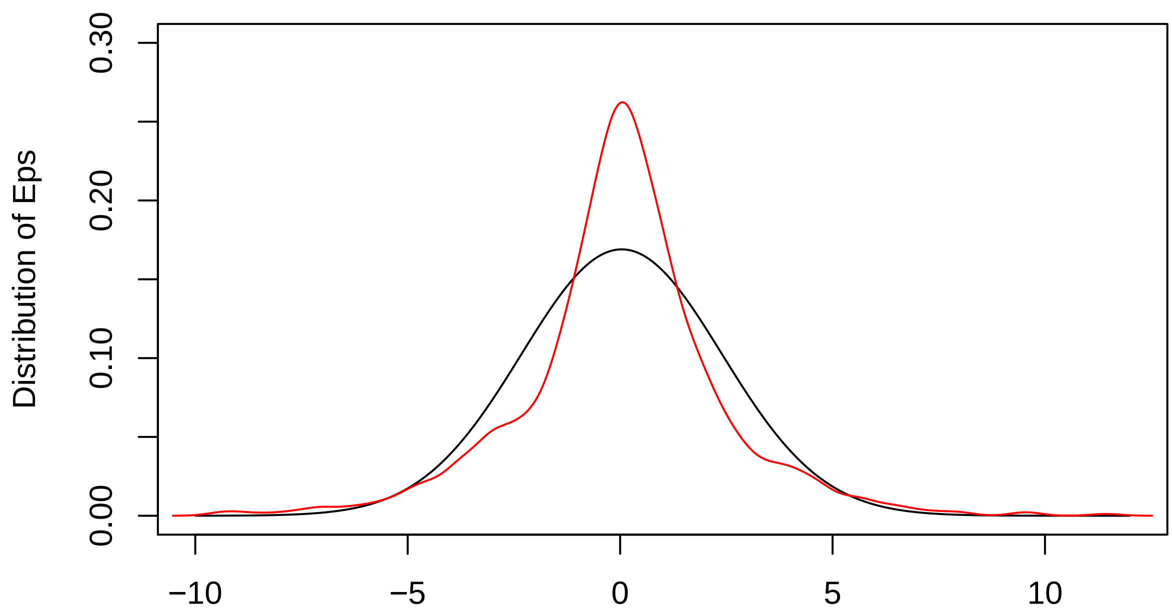
Simulate the linear model residuals Eps with changing standard deviations. And plot the residuals.

```
set.seed(1112131415);
Eps<-rnorm(nSample)*sd.process
plot(Eps,type="l")
```



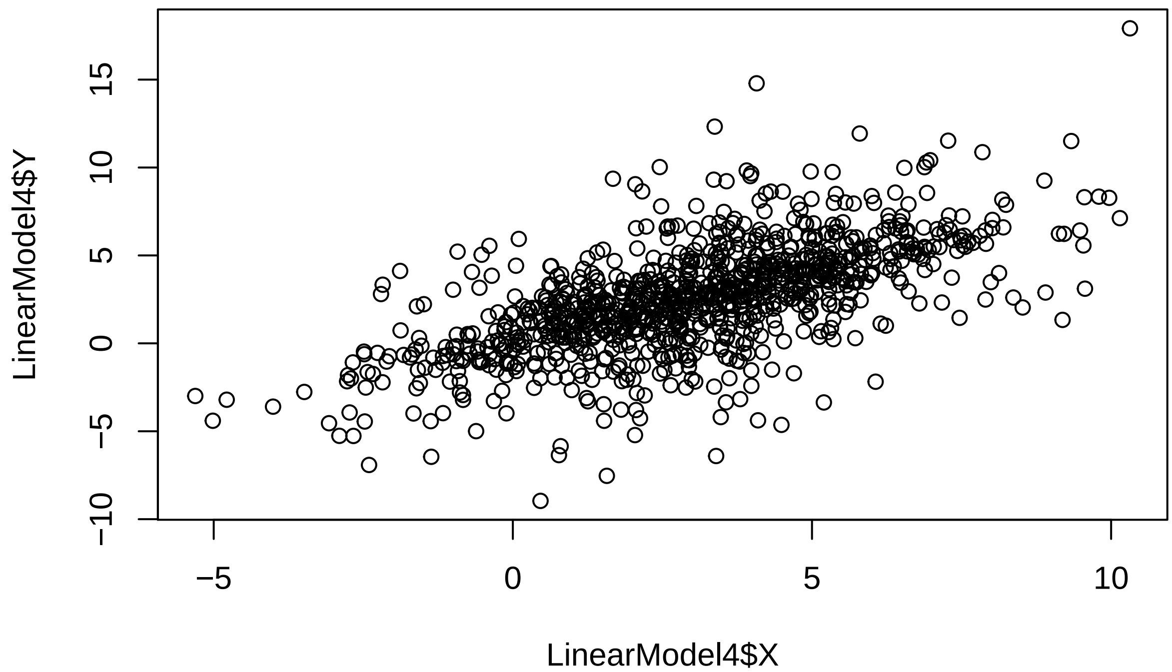
Observe how heteroscedasticity transforms normal distribution into leptokurtic distribution.

```
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")
```



Plot Linear Model 4

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



5. Effect of Residual Distribution on Correlation

Calculate the theoretical ??2 for the "correct model" which is LinearModel1.

```
set.seed(111)
X<-rnorm(nSample, 3, 2.5)
set.seed(1112131415)
Eps<-rnorm(nSample, 0, 1.5)
sd.X<-sd(X)
sd.Eps<-sd(Eps)
Theoretical.Rho.Squared<-(a*sd.X)^2/((a*sd.X)^2+sd.Eps^2)
Theoretical.Rho.Squared
```

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

And compare with the estimated ??^2 for each model:

```
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 Linear Model (Model 1) has the correlation closest to the theoretical correlation. If you distribute the errors differently, as you do in other models, it will change the observed correlation, sometimes drastically as

in the Cauchy distribution.

6. Estimation of Parameters

Estimate parameters a,b,?? using the function lm()

```
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"
```

```
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 the two estimates of the variance of the residuals:

```
var(summary(m1)$residuals)*999/998
```

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

Estimate the same parameters using the method of moments directly.

```
aEstimate<-cov(LinearModel1$X, LinearModel1$Y)/var(LinearModel1$X)
bEstimate<-mean(LinearModel1$Y)-(cov(LinearModel1$X, LinearModel1$Y)/var(LinearModel1$X))*mean(LinearMo
sigmaEstimate<-sqrt(var(LinearModel1$Y)-(cov(LinearModel1$X, LinearModel1$Y)/var(LinearModel1$X))^2*var
c(aEstimate, bEstimate, sigmaEstimate)
```

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

Reconcile sigmaEstimate with m1$sigma.

```
c(sigmaMetodMoments=sigmaEstimate,sigmaLinearModel=summary(m1)$sigma)
```

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

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.

```
m2<-lm(Y~X, data = LinearModel2)
m3<-lm(Y~X, data = LinearModel3)
m4<-lm(Y~X, data = LinearModel4)
```

```
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
```

```
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
```

```
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
```

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

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

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

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

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

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

Test

Download your sample, fit linear model to it, decide which distribution was used to simulate the residuals: normal, uniform, exponential, Cauchy.

Find:

Slope (10%) Intercept (10%) Mean value of residuals (10%) Standard deviation of residuals (30%) Distribution of residuals (40%)

```r
Path<-"C:/Users/mjdun/Desktop/Master Classes/Q1/Statistical Analysis/Lecture 3/"
df <- read.table(paste0(Path, 'Week3_Test_Sample.csv'), header=TRUE)
head(df)
```

```
##            Y         X
## 1 -32.105343  3.588052
## 2 -10.576851  2.173160
## 3   4.401168  2.220940
## 4  15.783844 -2.755864
## 5 -12.017638  2.572810
## 6  -8.011777  3.350696
```

```r
a<-cov(df$Y, df$X)/var(df$X)
a
```

```
## [1] 1.688955
```

```r
b<-mean(df$Y)-(cov(df$Y, df$X)/var(df$X))*mean(df$X)
b
```

```
## [1] 34.20783
```

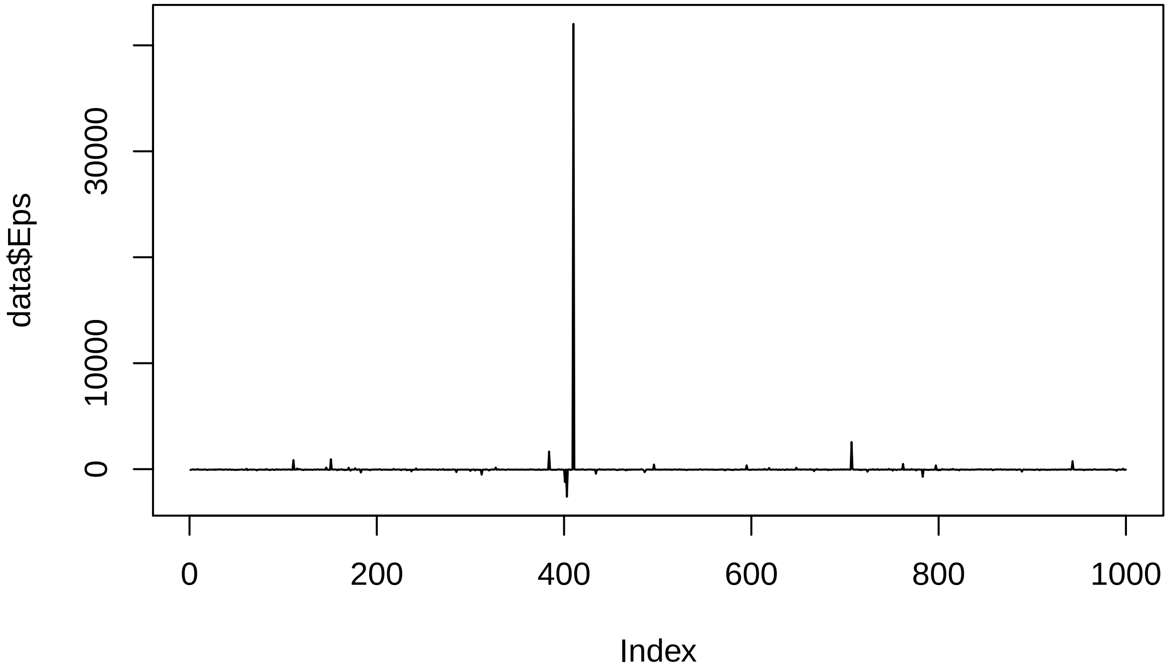
```r
Eps<-df$Y-a*df$X-b
data<-data.frame(Y=df$Y, X=df$X, Eps=Eps)
mean(data$Eps)
```

```
## [1] -4.976665e-15
```

```r
sd.Eps<-sd(data$Eps)
sd.Eps
```

```
## [1] 1338.848
```

```r
plot(data$Eps, type = "l")
```




```r
model<-lm(Y~X, data = df)
summary(model)$sigma
```

```
## [1] 1339.519
```