Generalized Linear Model

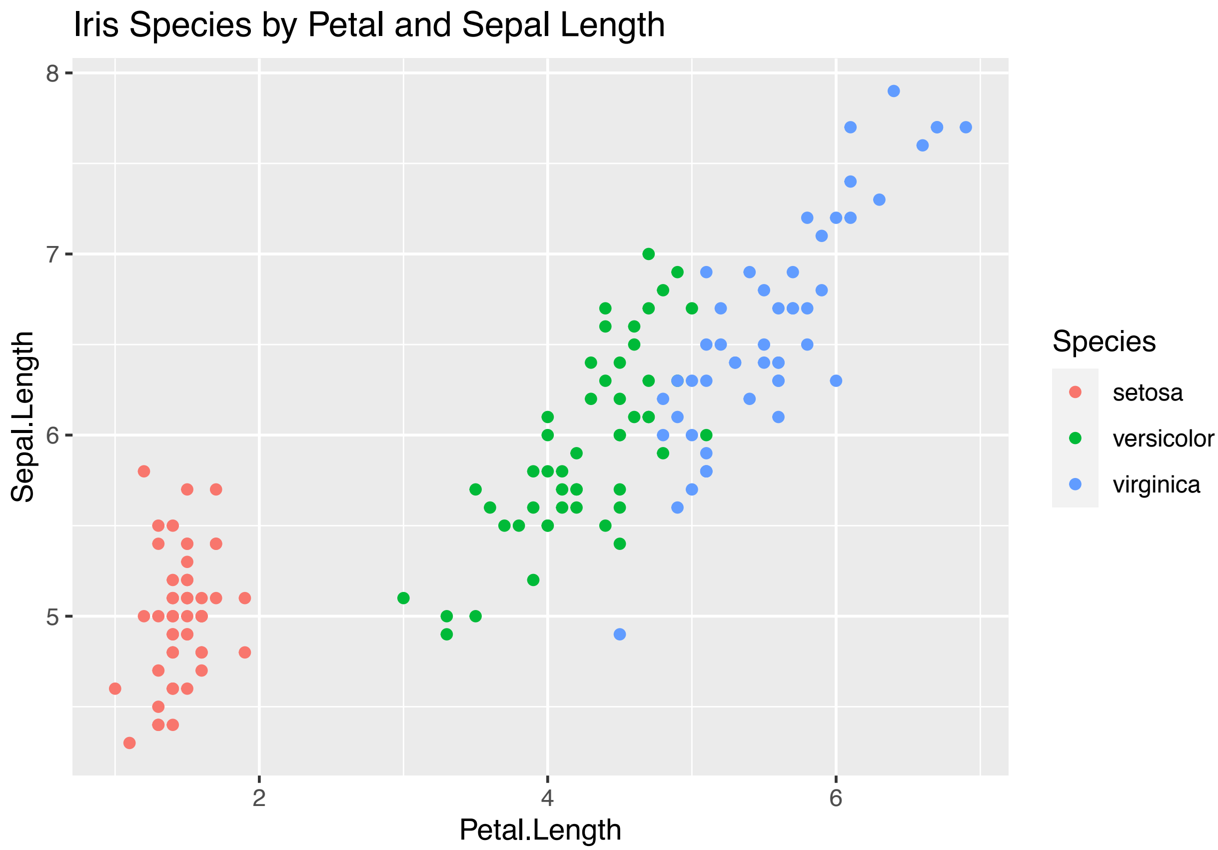
```r
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

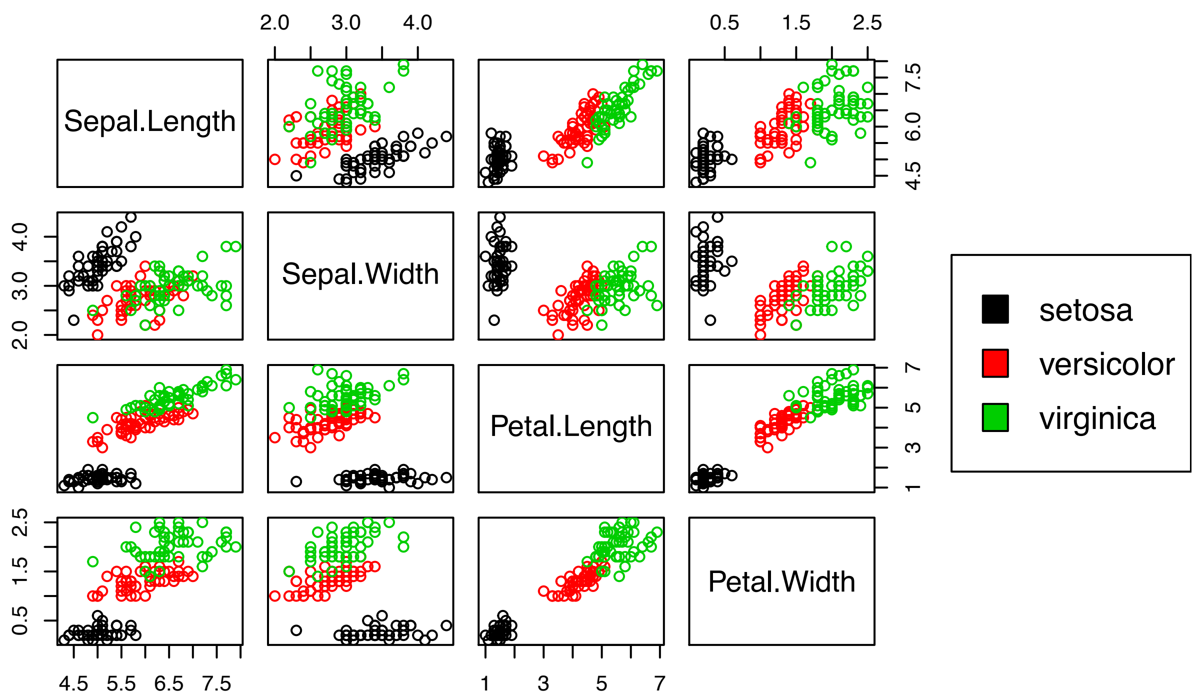
```r
library(ggplot2)
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2021a.
## 2.0/zoneinfo/Asia/Kolkata'
```

```r
ggplot(iris, aes(x = Petal.Length, y = Sepal.Length, colour = Species)) +
  geom_point() +
  ggtitle('Iris Species by Petal and Sepal Length')
```




```r
pairs(iris[,1:4],col=iris[,5],oma=c(4,4,6,12))
par(xpd=TRUE)
legend(0.85,0.6, as.vector(unique(iris$Species)),fill=c(1,2,3))
```



```r
iris[['Is.virginica']] <- as.numeric(iris[['Species']] == 'virginica')

head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species Is.virginica
## 1          5.1         3.5          1.4         0.2  setosa            0
## 2          4.9         3.0          1.4         0.2  setosa            0
## 3          4.7         3.2          1.3         0.2  setosa            0
## 4          4.6         3.1          1.5         0.2  setosa            0
## 5          5.0         3.6          1.4         0.2  setosa            0
## 6          5.4         3.9          1.7         0.4  setosa            0
```

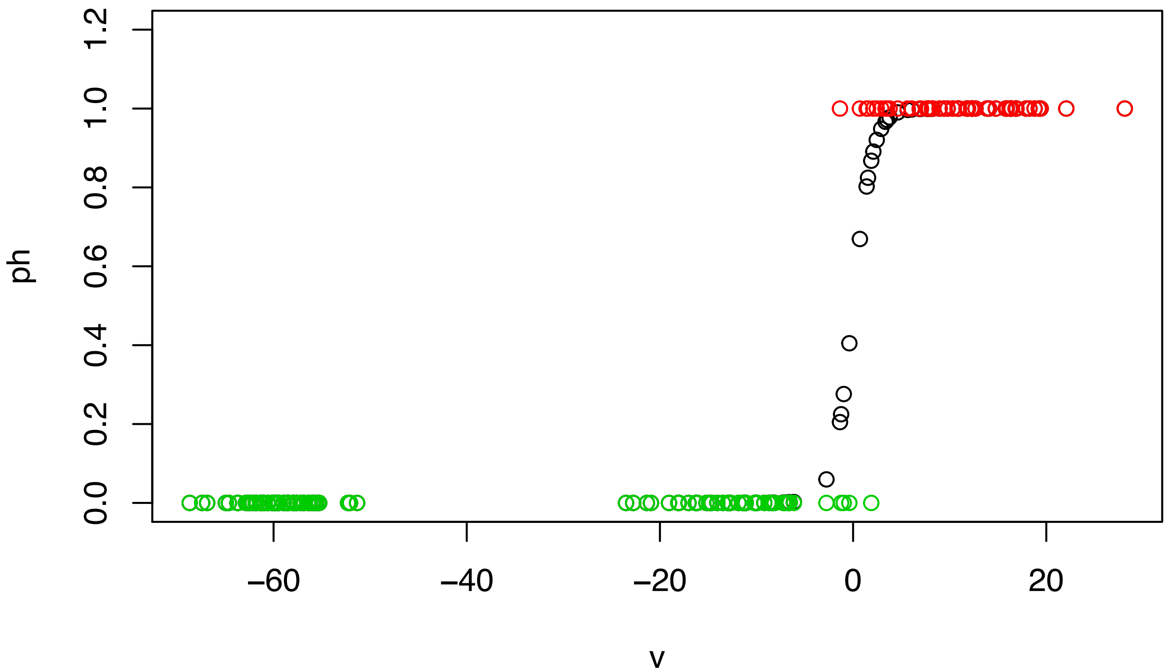
```r
fit.logit1 <- glm(Is.virginica ~ Petal.Length+Sepal.Length+Sepal.Width+Petal.Width, data
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(fit.logit1)
```

```
##
## Call:
## glm(formula = Is.virginica ~ Petal.Length + Sepal.Length + Sepal.Width +
##     Petal.Width, family = binomial(link = "logit"), data = iris)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.01105  -0.00065   0.00000   0.00048   1.78065
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -42.638     25.708  -1.659   0.0972 .
## Petal.Length    9.429      4.737   1.990   0.0465 *
## Sepal.Length   -2.465      2.394  -1.030   0.3032
## Sepal.Width    -6.681      4.480  -1.491   0.1359
## Petal.Width    18.286      9.743   1.877   0.0605 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 190.954  on 149  degrees of freedom
## Residual deviance:  11.899  on 145  degrees of freedom
## AIC: 21.899
##
## Number of Fisher Scoring iterations: 12
```

```
v<-predict(fit.logit1)
ph<-exp(v)/(1+exp(v))
par(mfrow=c(1,1))
plot(ph~v, ylim=c(0,1.2))
s0<-which(iris$Is.virginica==0)
s1<-which(iris$Is.virginica==1)
lines(iris$Is.virginica[s1]~v[s1], type = "p", col=2)
lines(iris$Is.virginica[s0]~v[s0], type = "p", col=3)
```




```
m<-min(ph[s1])
print(m)
```

```
## [1] 0.2048741
```

```
  iris[['Predict.virginica.logit']] <- as.numeric(predict(fit.logit1) > m)
table(iris[, c('Is.virginica', 'Predict.virginica.logit')])
```

```
##             Predict.virginica.logit
## Is.virginica  0  1
##            0 99  1
##            1  1 49
```

```
M<-max(ph[s0])
 print(M)
```

```
## [1] 0.8676299
```

```
  iris[['Predict.virginica.logit']] <- as.numeric(predict(fit.logit1) >M)
table(iris[, c('Is.virginica', 'Predict.virginica.logit')])
```

```
##              Predict.virginica.logit
## Is.virginica  0  1
##            0 99  1
##            1  2 48
```

- Three categories

```
library('nnet')

fit.logit2 <- multinom(Species~ Petal.Length+Sepal.Length+Sepal.Width+Petal.Width, data
```

```
## # weights:  18 (10 variable)
## initial  value 164.791843
## iter  10 value 16.177348
## iter  20 value 7.111438
## iter  30 value 6.182999
## iter  40 value 5.984028
## iter  50 value 5.961278
## iter  60 value 5.954900
## iter  70 value 5.951851
## iter  80 value 5.950343
## iter  90 value 5.949904
## iter 100 value 5.949867
## final  value 5.949867
## stopped after 100 iterations
```

```
predict_class<-predict(fit.logit2)
table(predict_class, iris$Species)
```

```
##
## predict_class setosa versicolor virginica
##    setosa         50          0         0
##    versicolor      0         49         1
##    virginica       0          1        49
```

```
summary (fit.logit2)
```

```
## Call:
## multinom(formula = Species ~ Petal.Length + Sepal.Length + Sepal.Width +
##     Petal.Width, data = iris)
##
## Coefficients:
##            (Intercept) Petal.Length Sepal.Length Sepal.Width Petal.Width
## versicolor    18.69037     14.24477    -5.458424   -8.707401   -3.097684
## virginica    -23.83628     23.65978    -7.923634  -15.370769   15.135301
##
## Std. Errors:
##            (Intercept) Petal.Length Sepal.Length Sepal.Width Petal.Width
## versicolor    34.97116     60.19170     89.89215    157.0415    45.48852
## virginica     35.76649     60.46753     89.91153    157.1196    45.93406
##
## Residual Deviance: 11.89973
```

TRANSFORMATION OF VARIABLES

Reasons for Making Transformations

(1) Remedies for non-normality
(2) Heterogeneous variances of the errors
(3) Simplify the relationship between the dependent variable and the independent variables.

Exponential growth curve

Model $y = \beta_0 x^{\beta_1} \upsilon$
Transformation $Y = \ln y$, $X = \ln x$, $\epsilon = \ln \upsilon$
Transformed model $Y = (\ln \beta_0) + \beta_1 X + \epsilon$

```
n<-200
bt<-c(2,4)
ep<-exp(rnorm(n))
m<-1 ; M<-5
x<- sort(runif(n,min = m,max = M))
z<-seq(m,M, by=0.01)
y<-bt[1]*x^(bt[2])*ep
yz<-bt[1]*z^(bt[2])

X<-log(x)
Y<-log(y)
Z<-log(z)
YZ<-log(yz)
# Data fitting
fit<-lm(Y~X)
beta_0_hat<-exp(fit$coefficient[1])
beta_1_hat<-(fit$coefficient[2])
cat("True beta_0=", (bt[1]), "estimated beta_0=" , beta_0_hat, "\n")
```

```
## True beta_0= 2 estimated beta_0= 1.750557
```

```
cat("True beta_1=", (bt[2]), "estimated beta_1=" , beta_1_hat, "\n")
```

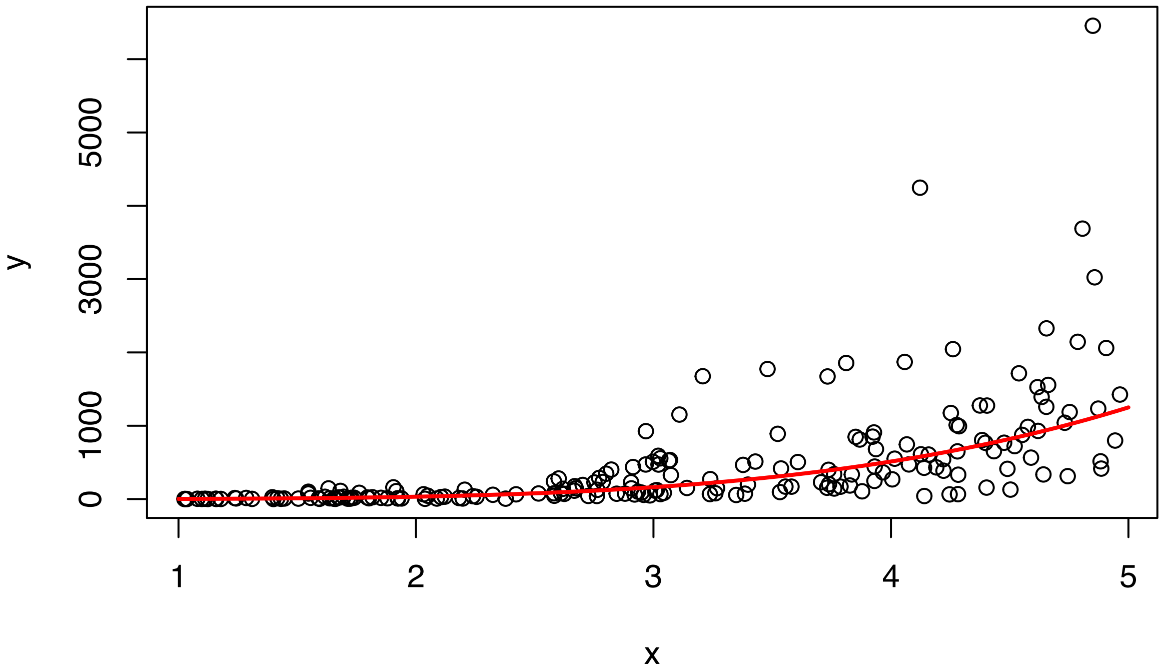
```
## True beta_1= 4 estimated beta_1= 4.099917
```

```
summary(fit)
```

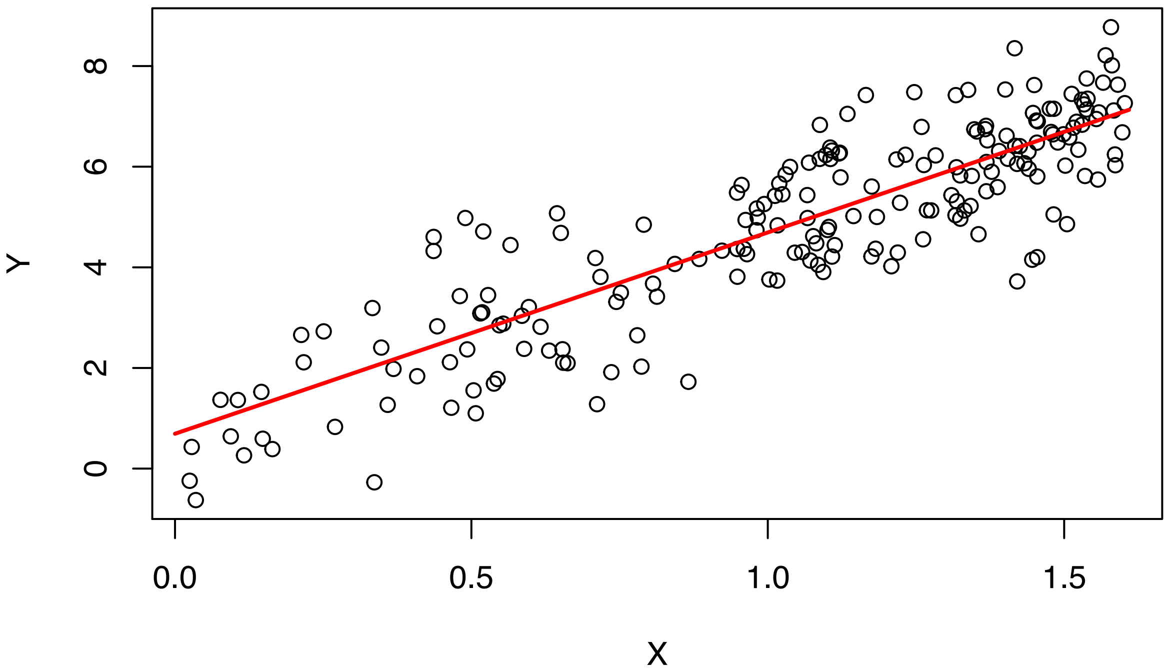
```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.66530 -0.65969  0.00288  0.62326  2.41266
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5599     0.1810   3.094  0.00226 **
## X             4.0999     0.1610  25.468  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9763 on 198 degrees of freedom
## Multiple R-squared:  0.7661, Adjusted R-squared:  0.7649
## F-statistic: 648.6 on 1 and 198 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(1,1))
plot(y~x)
lines(yz~z, col=2, lwd=2)
```




```r
plot(Y~X)
lines(YZ~Z, col=2, lwd=2)
```


Y
X
0
2
4
6
8
0.0
0.5
1.0
1.5

Exponential decay curve

Model $y = \beta_0 e^{x\beta_1} \upsilon$
Transformation $Y = \ln y$, $X = x$, $\epsilon = \ln \upsilon$
Transformed model $Y = (\ln \beta_0) + \beta_1 X + \epsilon$

```
n<-200
bt<-c(3,-1.4)
ep<-exp(rnorm(n))
m<-1 ;  M<-5
x<- sort(runif(n,min = m,max = M))
z<-seq(m,M, by=0.01)
y<-bt[1]*exp(x*(bt[2]))*ep
yz<-bt[1]*exp(z*(bt[2]))

X<-(x)
Y<-log(y)
Z<-(z)
YZ<-log(yz)
# Data fitting
fit<-lm(Y~X)
beta_0_hat<-exp(fit$coefficient[1])
beta_1_hat<-(fit$coefficient[2])
cat("True beta_0=", (bt[1]), "estimated beta_0=" , beta_0_hat, "\n")
```

```
## True beta_0= 3 estimated beta_0= 2.691813
```

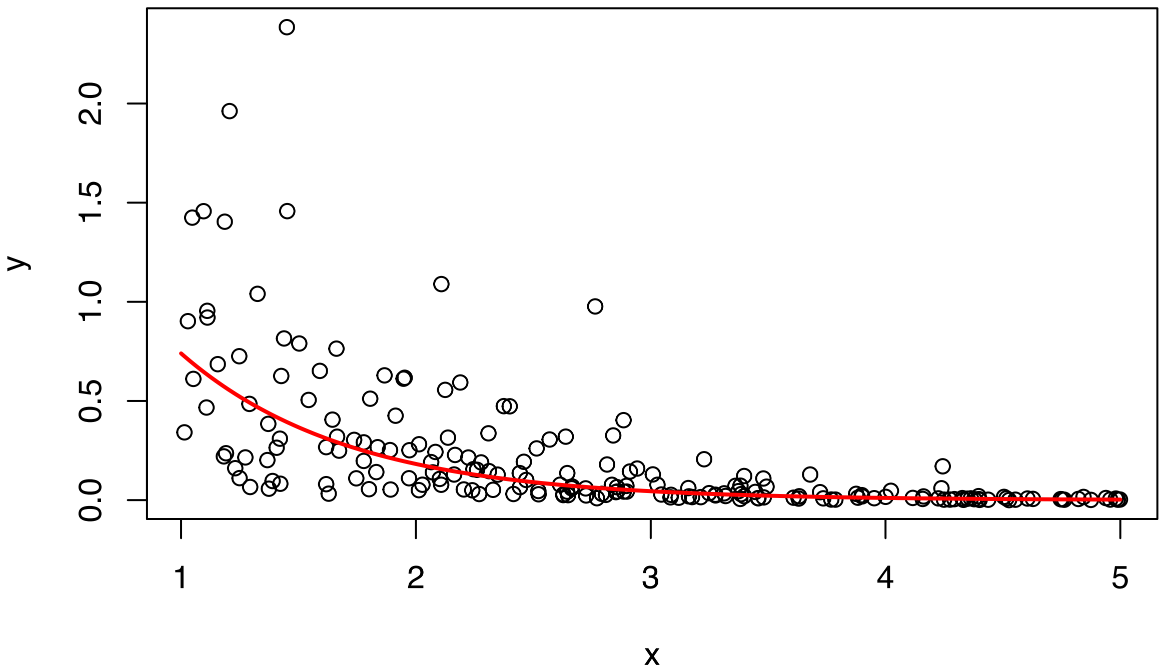
```
cat("True beta_1=", (bt[2]), "estimated beta_1=" , beta_1_hat, "\n")
```

```
## True beta_1= -1.4 estimated beta_1= -1.346354
```

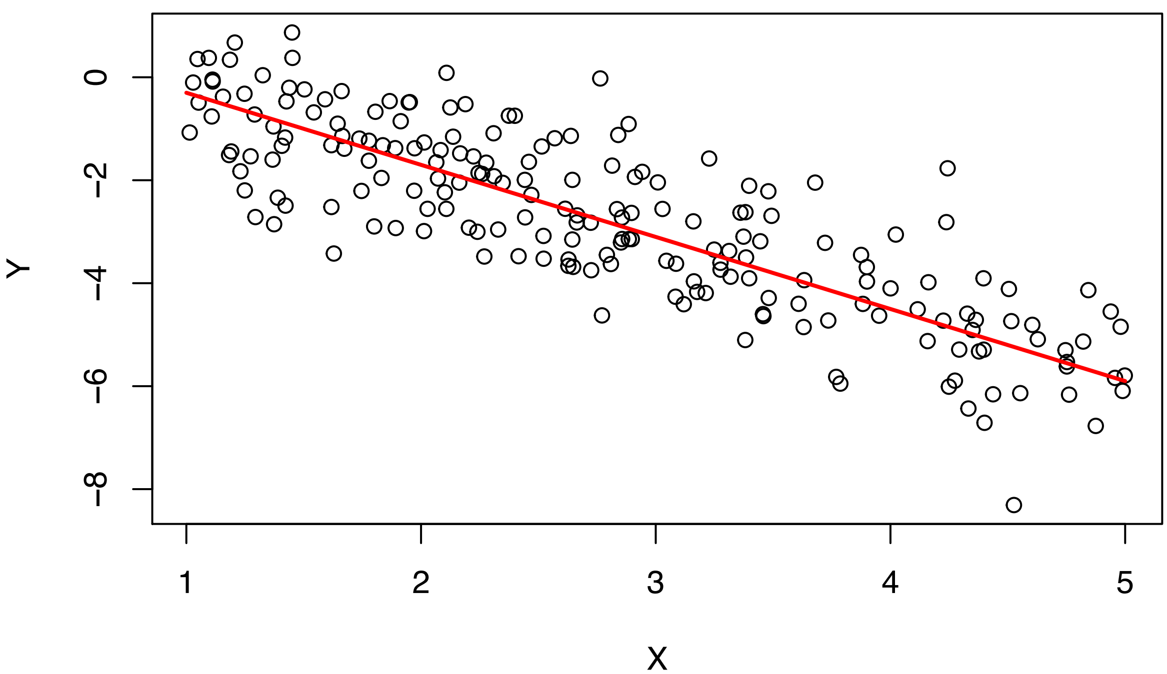
```
summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2053 -0.6969  0.0306  0.6895  2.9550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.99022    0.18674   5.303 3.04e-07 ***
## X           -1.34635    0.06231 -21.608  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9807 on 198 degrees of freedom
## Multiple R-squared:  0.7022, Adjusted R-squared:  0.7007
## F-statistic: 466.9 on 1 and 198 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,1))
plot(y~x)
lines(yz~z, col=2, lwd=2, ylim=c(0,5))
```




```r
plot(Y~X)
lines(YZ~Z, col=2, lwd=2)
```



Inverse polynomial model

Model $y = \frac{x}{\alpha_0+\alpha_1 x+\upsilon}$
Transformation $Y = 1/y$, $X = 1/x$, $\epsilon = \upsilon/x$
Transformed model $Y = (\beta_0) + \beta_1 X + \epsilon$ where $\beta_0 = \alpha_1$, $\beta_1 = \alpha_0$

```
n<-500
a<-c(30,4)
bt<-c(a[2],a[1])
ep<-(rnorm(n))
m<-1 ;  M<-5
x<- sort(runif(n,min = m,max = M))
z<-seq(m,M, by=0.01)
y<-x/(a[1]+a[2]*x+ ep*x)
yz<-z/(a[1]+a[2]*z)

X<-(1/x)
Y<-(1/y)
Z<-(1/z)
YZ<-(1/yz)
# Data fitting
fit<-lm(Y~X)
beta_0_hat<-(fit$coefficient[1])
beta_1_hat<-(fit$coefficient[2])
cat("True beta_0=", (bt[1]), "estimated beta_0=" , beta_0_hat, "\n")
```

```
## True beta_0= 4 estimated beta_0= 3.936817
```

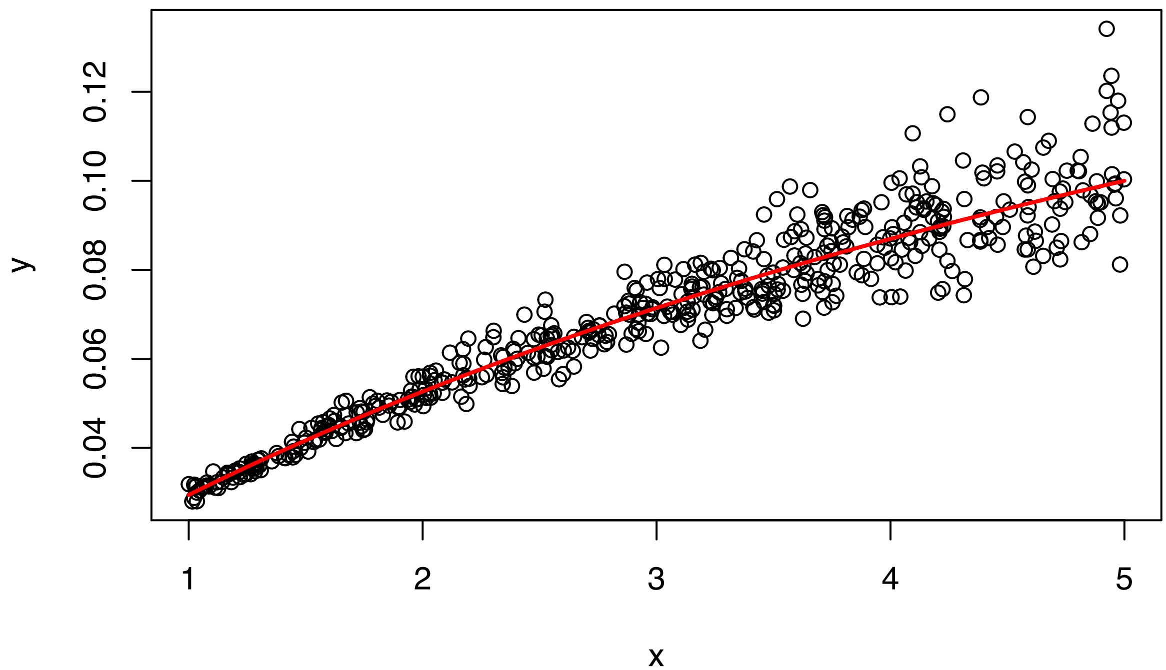
```
cat("True beta_1=", (bt[2]), "estimated beta_1=" , beta_1_hat, "\n")
```

```
## True beta_1= 30 estimated beta_1= 30.26974
```

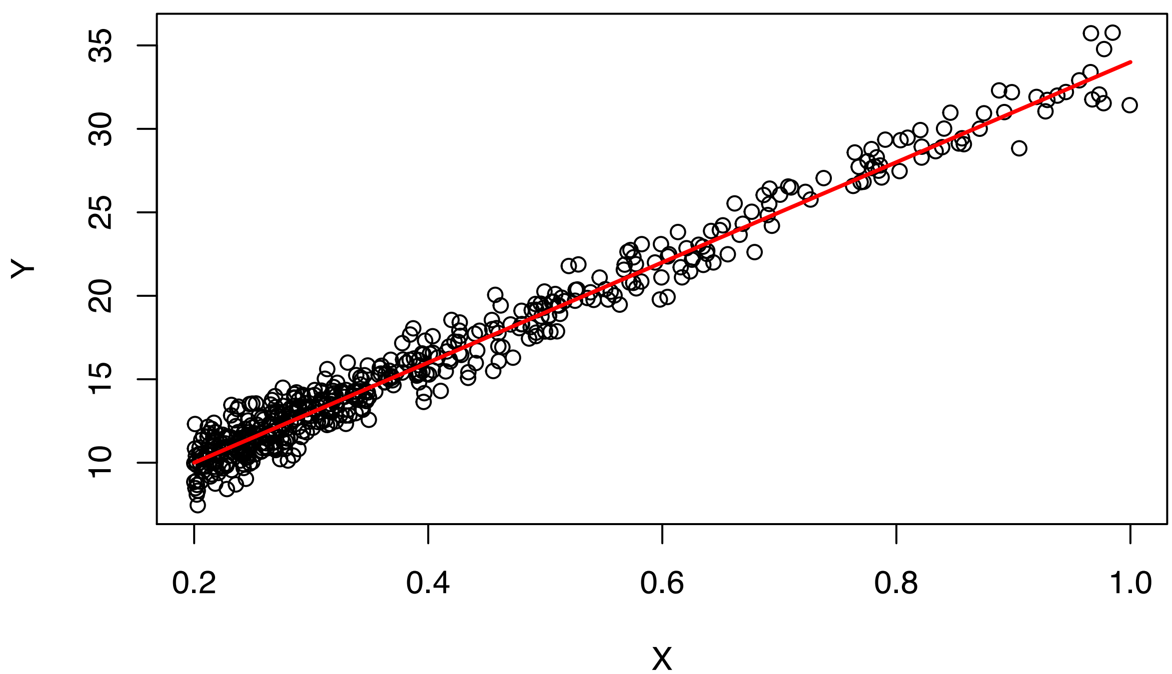
```
summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.77024 -0.64909  0.03161  0.64621  2.54088
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.9368     0.1016   38.76   <2e-16 ***
## X            30.2697     0.2270  133.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.001 on 498 degrees of freedom
## Multiple R-squared:  0.9728, Adjusted R-squared:  0.9727
## F-statistic: 1.778e+04 on 1 and 498 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,1))
plot(y~x)
lines(yz~z, col=2, lwd=2, ylim=c(0,5))
```




```
plot(Y~X)
lines(YZ~Z, col=2, lwd=2)
```



Logistic Growth Model

Model $y = \frac{1}{1+\alpha_0 e^{\alpha_1 x} \upsilon}$
Transformation $Y = \log\left(\frac{1}{y} - 1\right)$, $X = x$, $\epsilon = \log \upsilon$
Transformed model $Y = (\beta_0) + \beta_1 X + \epsilon$ where $\beta_0 = \log \alpha_0$, $\beta_1 = \alpha_1$

```r
n<-200
a<-c(2,-1.4)
bt<-c(log(a[1]),a[2])
ep<-exp(rnorm(n))
m<- -5 ; M<-5
x<- sort(runif(n,min = m,max = M))
z<-seq(m,M, by=0.01)
y<-1/(1+a[1]*exp(a[2]*x+ep))
yz<-1/(1+a[1]*exp(a[2]*z))

X<-(x)
Y<-log(1/y-1)
Z<-(z)
YZ<- (1+a[1]+a[2]*(Z))
# Data fitting
fit<-lm(Y~X)
beta_0_hat<-(fit$coefficient[1])
beta_1_hat<-(fit$coefficient[2])
cat("True alpha_0=", (a[1]), "estimated alpha_0=" , beta_0_hat, "\n")
```

```
## True alpha_0= 2 estimated alpha_0= 2.62309
```

```r
cat("True alpha_1=", (a[2]), "estimated alpha_1=" , beta_1_hat, "\n")
```

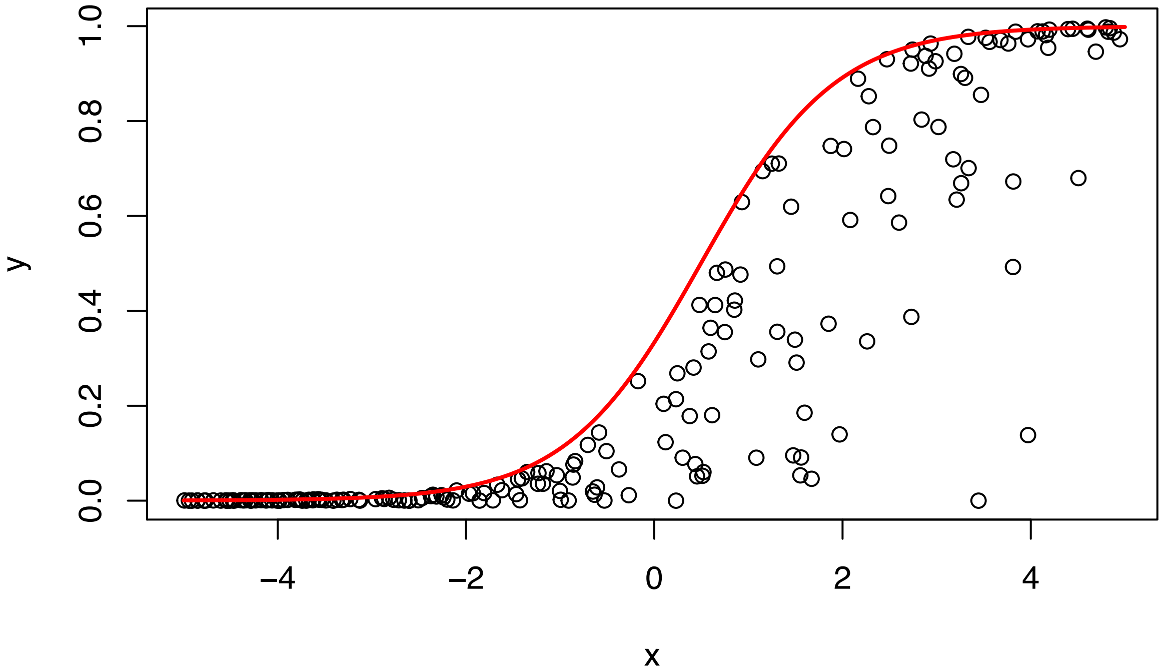
```
## True alpha_1= -1.4 estimated alpha_1= -1.344172
```

```r
summary(fit)
```

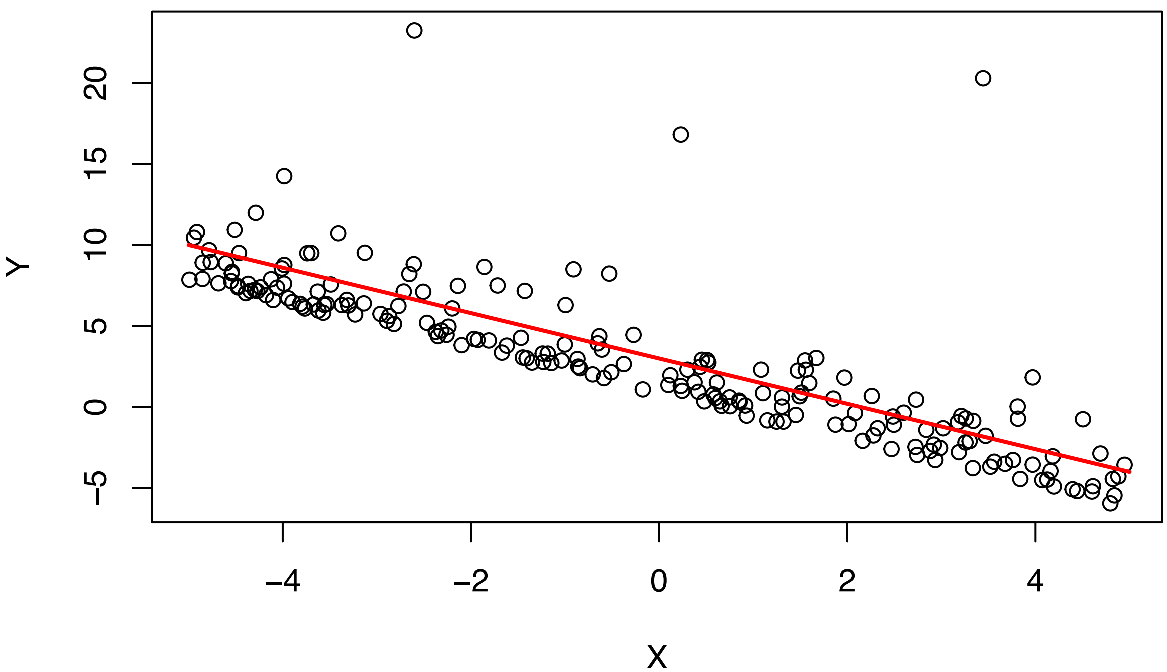
```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1207 -1.2922 -0.8481  0.5287 22.3013
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.62309    0.19361   13.55   <2e-16 ***
## X           -1.34417    0.06472  -20.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.72 on 198 degrees of freedom
## Multiple R-squared:  0.6854, Adjusted R-squared:  0.6838
## F-statistic: 431.3 on 1 and 198 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(1,1))
plot(y~x)
```

```r
lines(yz~z, col=2, lwd=2)
```




```r
plot(Y~X)
lines(YZ~Z, col=2, lwd=2)
```



Variance Stabelizing Transformation

CASE 1 : $\sigma^2 \propto constant$

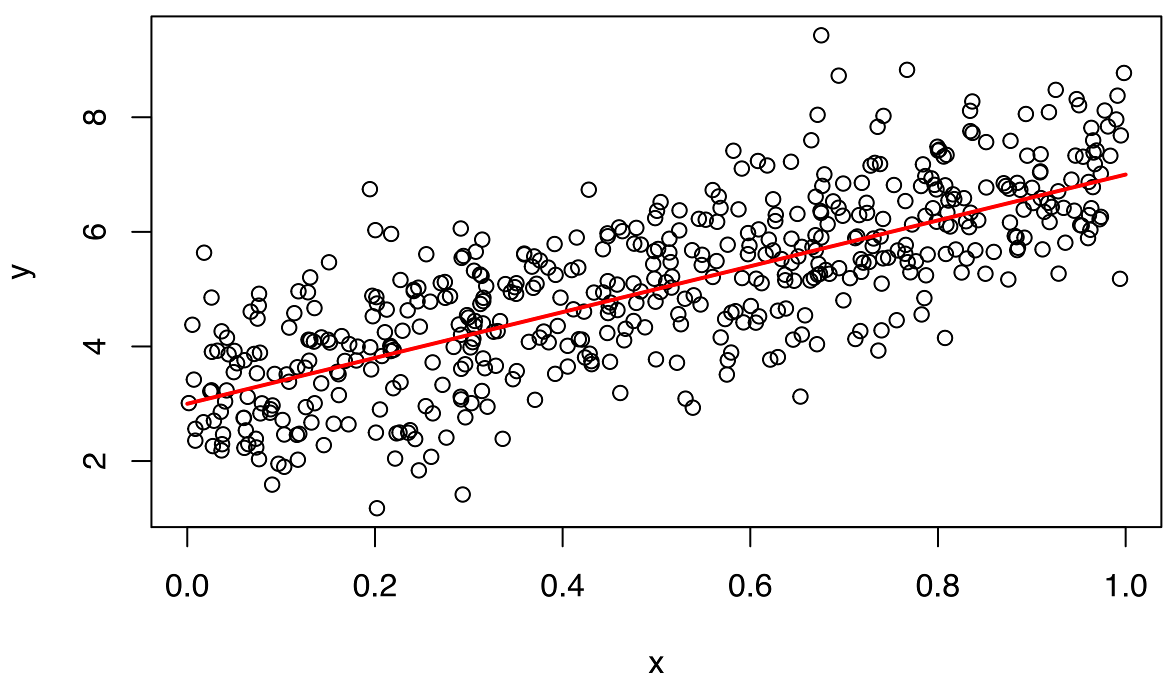
```
n<-500
bt<-c(3,4)
ep<-(rnorm(n))
m<-0 ;  M<-1
x<- sort(runif(n,min = m,max = M))
z<-seq(m,M, by=0.01)
y<-bt[1]+bt[2]*x+ep
yz<- bt[1]+bt[2]*z
# Data fitting
fit<-lm(y~x)
beta_0_hat<-(fit$coefficient[1])
beta_1_hat<-(fit$coefficient[2])
cat("True beta_0=", (bt[1]), "estimated beta_0=" , beta_0_hat, "\n")
```

```
## True beta_0= 3 estimated beta_0= 3.029625
```

```
cat("True beta_1=", (bt[2]), "estimated beta_1=" , beta_1_hat, "\n")
```

```
## True beta_1= 4 estimated beta_1= 4.096683
```

```
#summary(fit)
par(mfrow=c(1,1))
plot(y~x)
lines(yz~z, lwd=2, col=2)
```

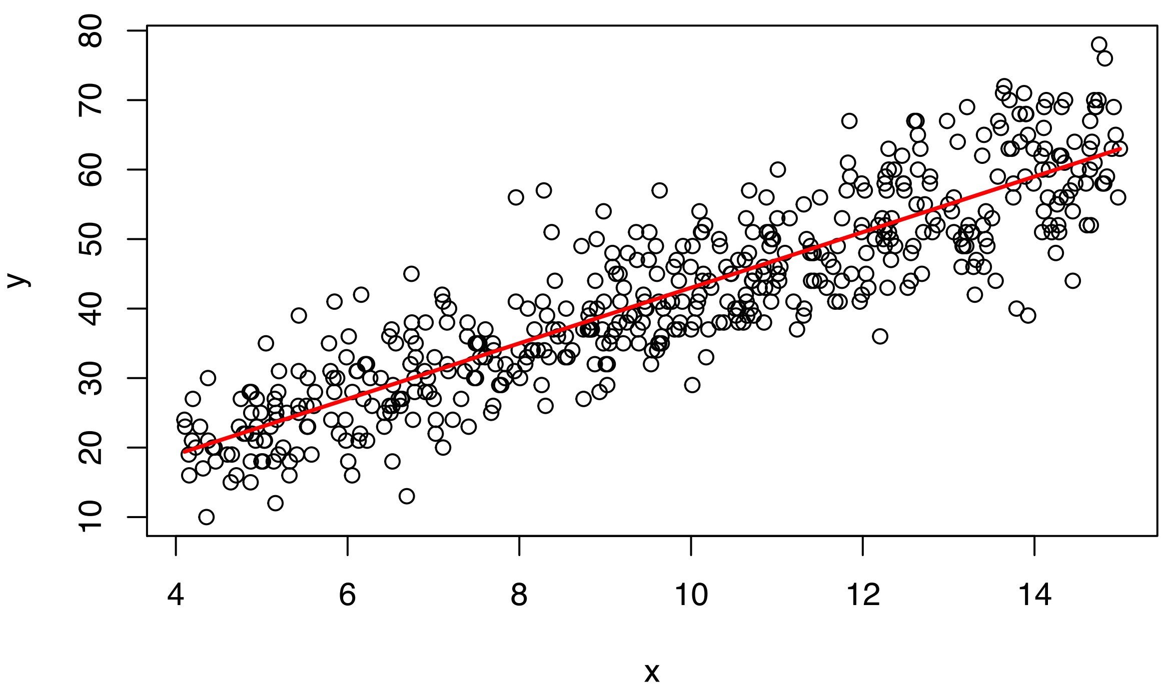


CASE 2 : $\sigma^2 \propto E(y)$ **Tranform** $Y = \sqrt(y)$

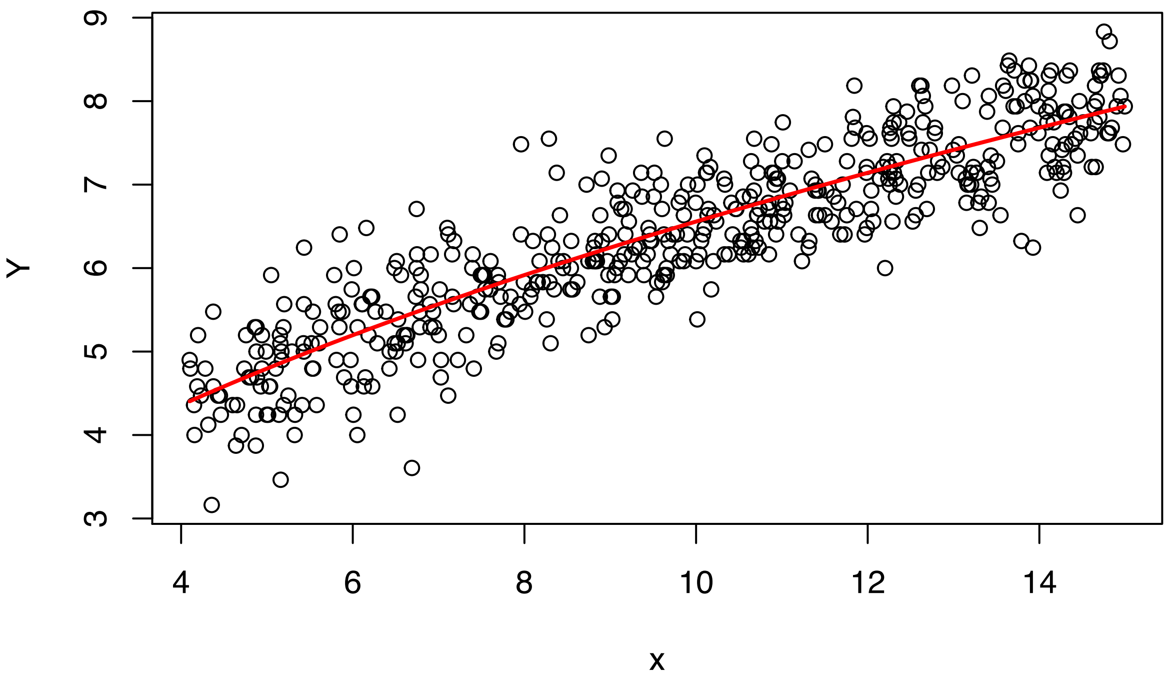
```
n<-500
bt<-c(3,4)

m<-4 ;  M<-15
x<- sort(runif(n,min = m,max = M))
z<-seq(m,M, by=0.01)
y<-numeric(0)
for(i in 1 : n){
  a<-bt[1]+bt[2]*x[i]
  y[i]<-rpois(1,a)
}
Y<-sqrt(y)

# Data fitting
fit<-lm(Y~x)
print(fit$coefficients)
```

```
## (Intercept)           x
##   3.2725397   0.3182228
```

```
plot(y~x)
lines((bt[1]+bt[2]*x)~x, col=2, lwd=2)
```




```
plot(Y~x)
lines(sqrt(bt[1]+bt[2]*x)~x, col=2, lwd=2)
```

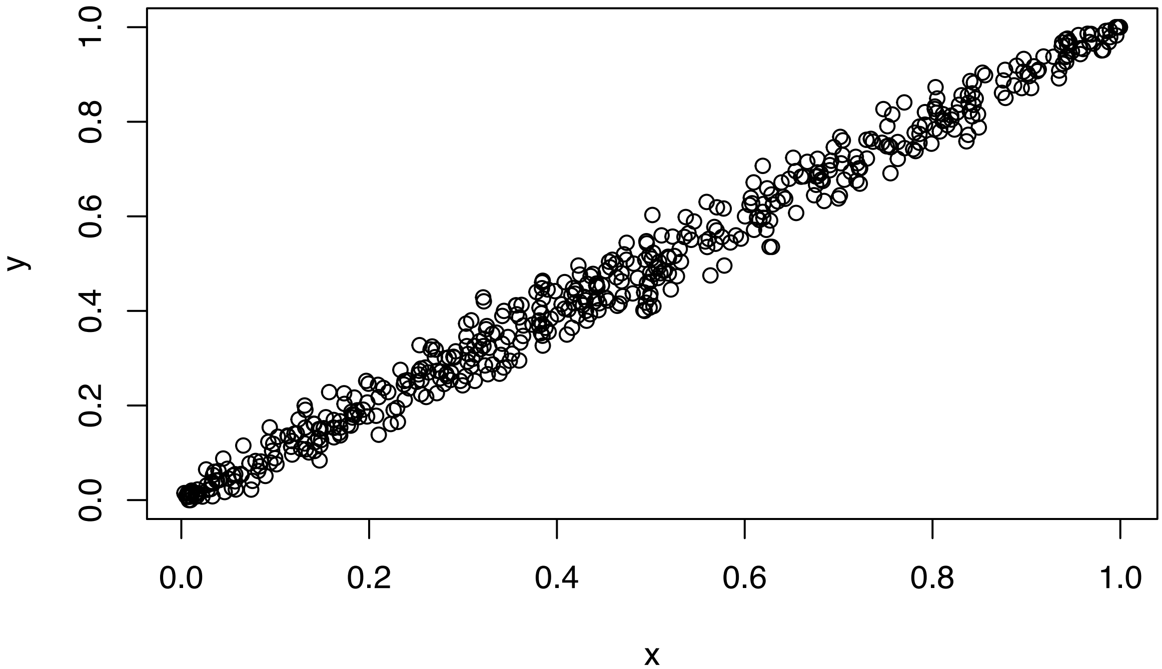


CASE 3 : $\sigma^2 \propto E(y)(1-E(Y))$ **Tranform** $Y = sin^{-1}(\sqrt{(}y))$

```
n<-500
m<- 50+sort(rpois(n,80))
x<-runif(n,0,1)
y<-numeric(0)
for(i in 1 : n){
  y[i]<-(rbinom(1,m[i],x[i])/m[i])
}
Y<-asin(sqrt(y))

# Data fitting
fit<-lm(Y~x)
print(fit$coefficients)
```

```
## (Intercept)           x
##   0.1844797   1.2095198
```

```
plot(y~x)
```

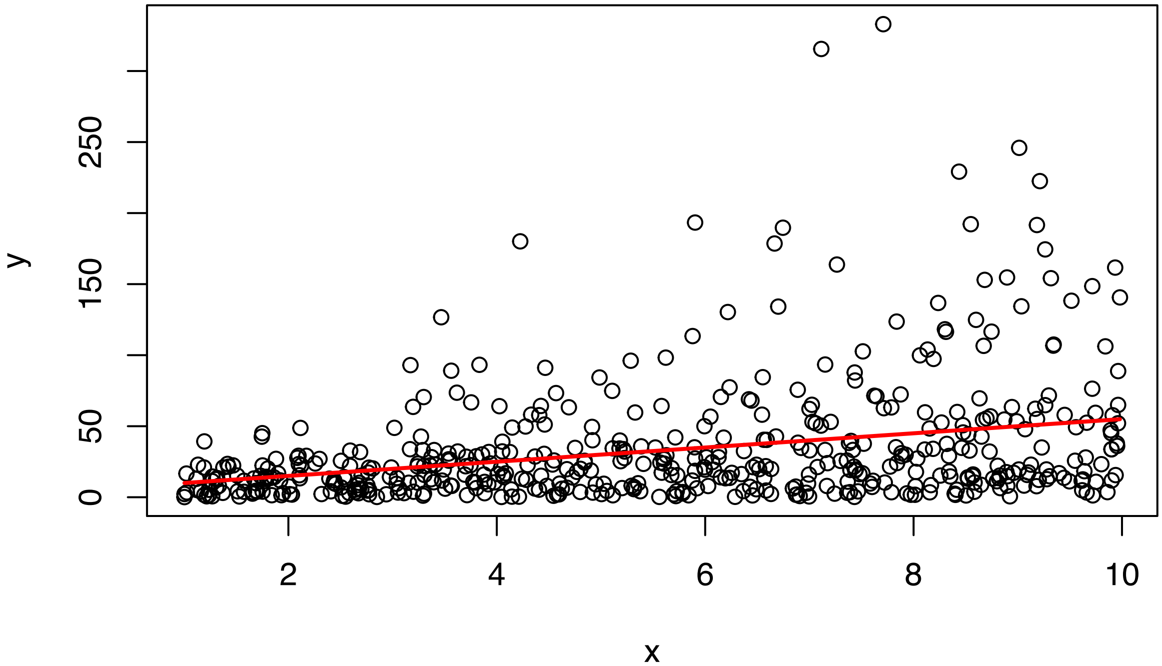


CASE 4 : $\sigma^2 \propto (E(y))^2$ **Tranform** $Y = log(y)$

```
n<-500
bt<-c(5,5)

m<-1;  M<-10
x<- sort(runif(n,min = m,max = M))
z<-seq(m,M, by=0.01)
y<-numeric(0)
for(i in 1 : n){
  a<-bt[1]+bt[2]*x[i]
  y[i]<-rexp(1,1)*a    #rgamma(1, shape=a,rate=1)
}
Y<-log(y)

# Data fitting
fit<-lm(Y~x)
print(fit$coefficients)
```

```
## (Intercept)           x
##   1.8850788   0.1667183
```

```
plot(y~x)
lines((bt[1]+bt[2]*x)~(x), col=2, lwd=2)
```




```r
plot(Y~x)
```