dataTransformations

September 26, 2019

0.1 data transformations

The regression function we have explored take the form,

$$p(y|X) \sim N(X\beta, \sigma^2),$$

and this model makes some important assumptions. * Our response (y) is linearly related to X. * The observations (y_i, x_i) are independent from one another. * The conditional probability of our response y is normally distributed. * The same σ applies to all values of X. This is called homoskedasticity of errors.

0.2 when these assumptions fail

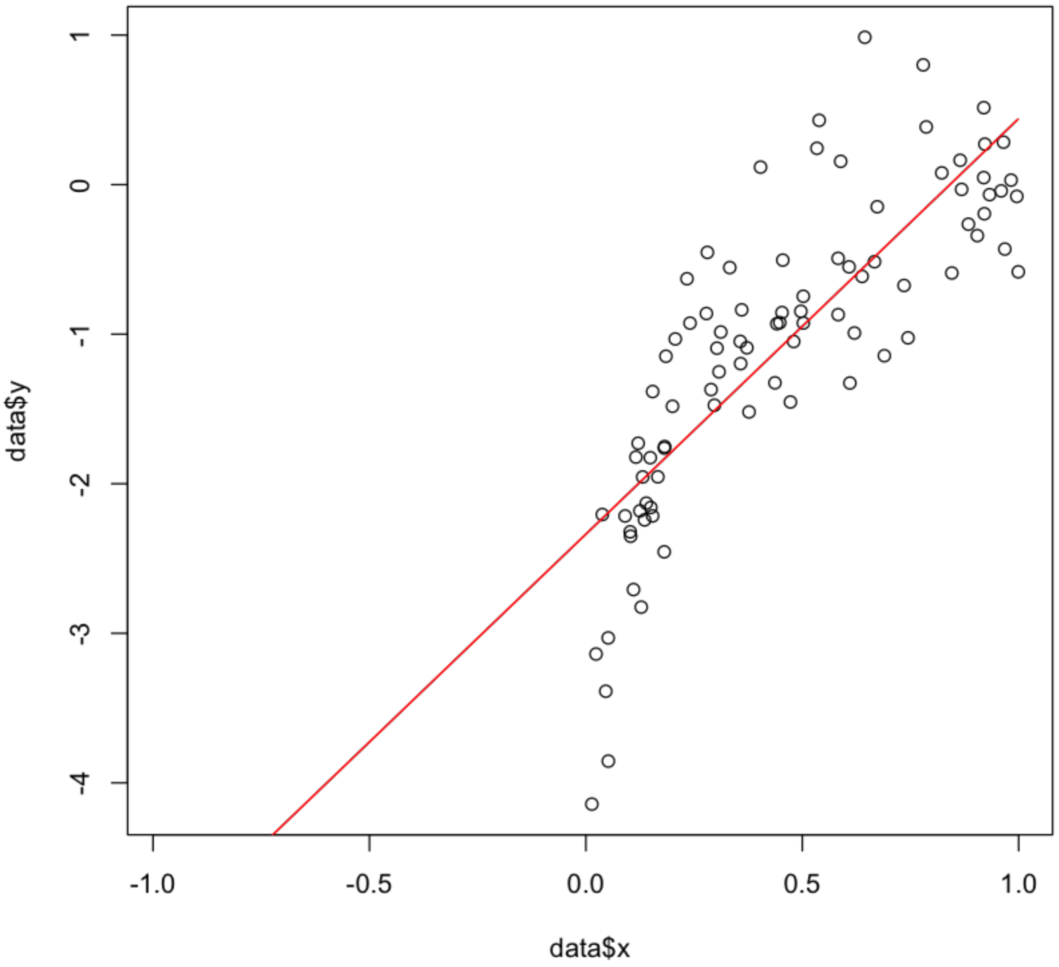
When our collected data fails to fit all the assumptions of linear regression we have two options: (i) pick a different, more complicated model or (ii) transform our original data so that our transformed data meet the above linear regression assumptions. We'll spend quite a bit of time on (i). For now, lets spend time on (ii).

0.2.1 Exploratory Data Analysis

X and Y data in hand, we can plot the relationship between these variables and a linear regression fit.

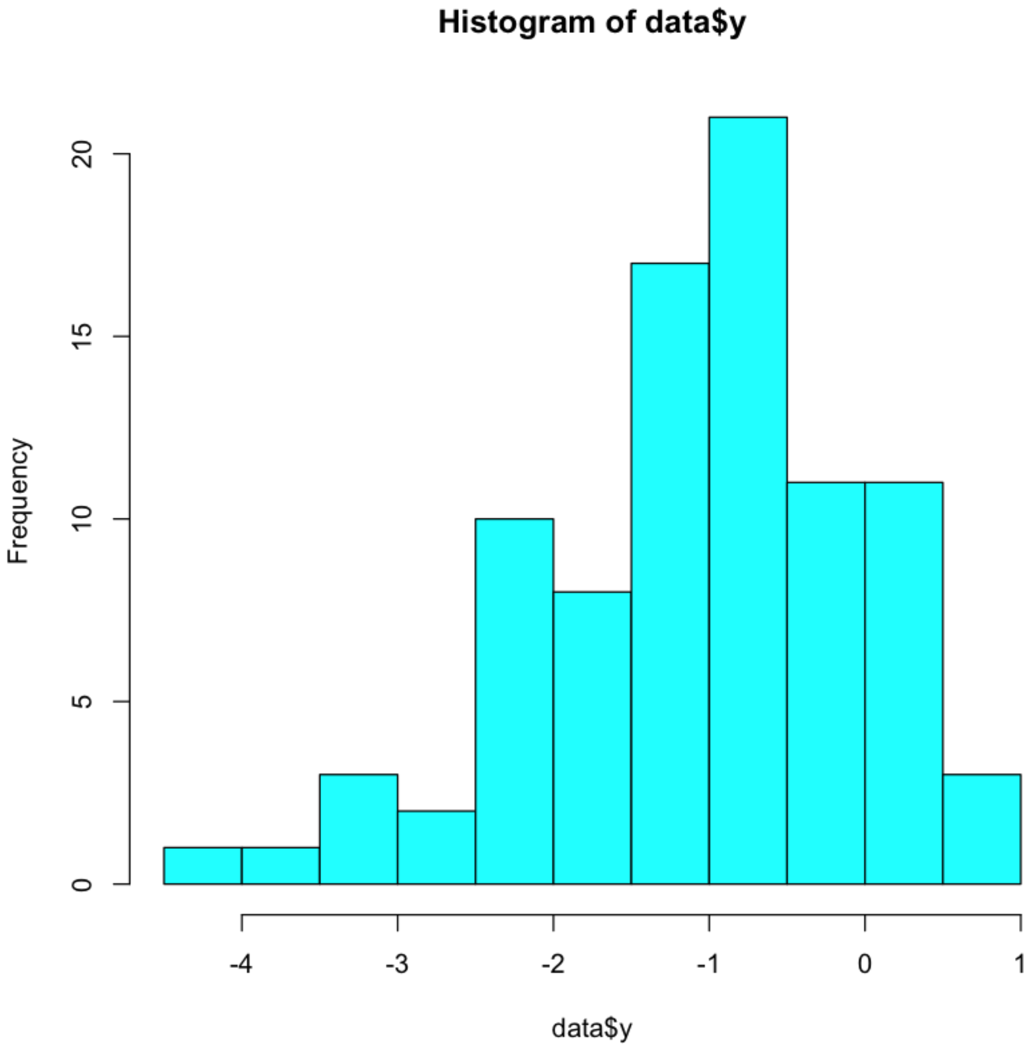
[69]:

```r
data <- read.csv("./dataSet1.csv")
plot(data$x,data$y)
model1 <- lm(y~x,data=data)
lines(data$x,predict.lm(model1,data.frame(x=data$x)),col='red')
```



The linear regression fits well for x values great than 1/2 but it looks like the relationship between x and y is non-linear. Before we begin transforming our variables, let look at a histogram of y.

[70]:
```
hist(data$y,10,col='cyan')
```



0.3 Checking linearity (transforming X)

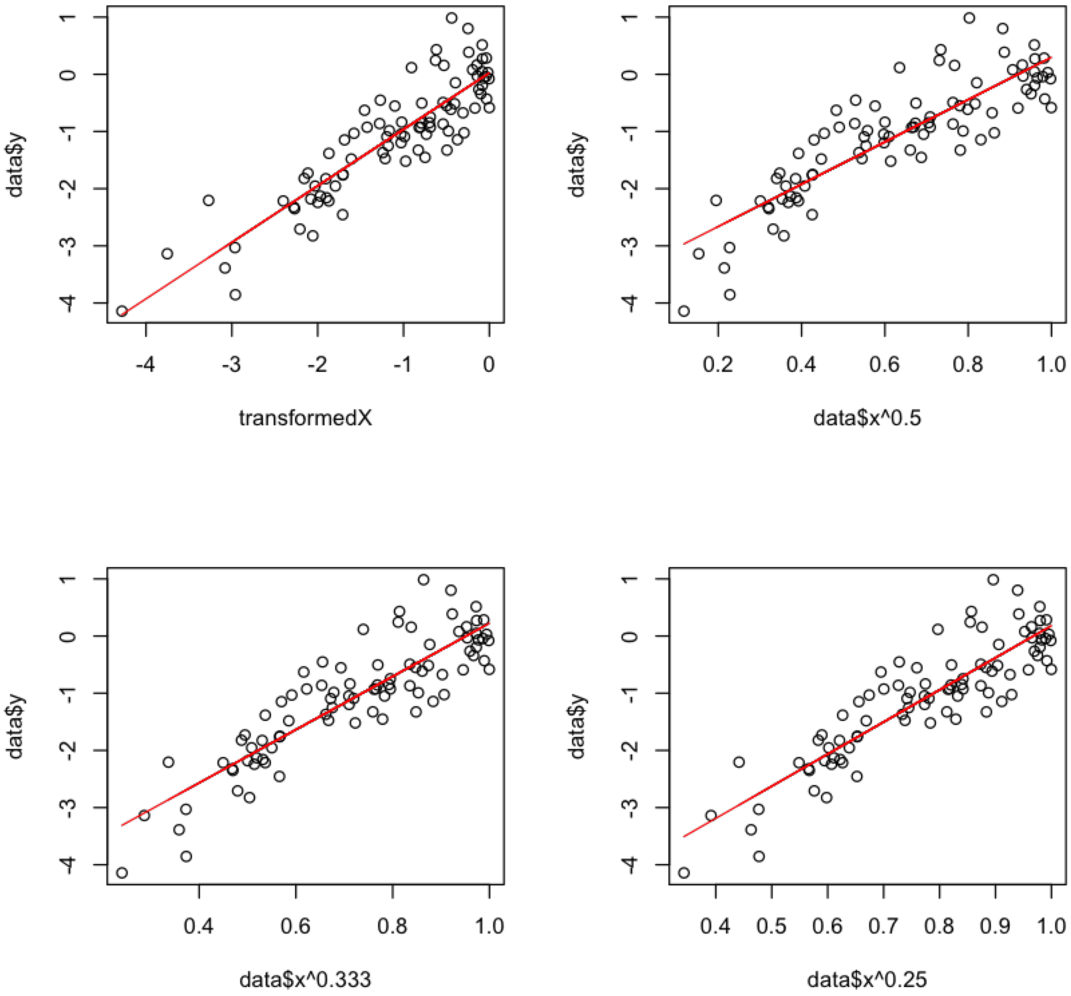
Our y values seem to exhibit a roughly Normal shape. Transforming the y covariate may move the distribution of y further away from normal. We can look at different transformation of x first.

[60]:

```
par(mfrow=c(2,2))

transformedX = log(data$x)
plot(transformedX,data$y)
model <- lm(y~log(x),data=data)

predictions <- predict.lm(model,data.frame(x=data$x))
```

```
lines(transformedX
      ,predictions
      ,col='red')

transformedX = (data$x)^0.5
plot(data$x^0.5,data$y)
model <- lm(y~I(x^0.5),data=data)

predictions <- predict.lm(model,data.frame(x=data$x))
lines(transformedX
      ,predictions
      ,col='red')

transformedX = (data$x)^0.333
plot(data$x^0.333,data$y)
model <- lm(y~I(x^0.333),data=data)

predictions <- predict.lm(model,data.frame(x=data$x))
lines(transformedX
      ,predictions
      ,col='red')

transformedX = (data$x)^0.25
plot(data$x^0.25,data$y)
model <- lm(y~I(x^0.25),data=data)

predictions <- predict.lm(model,data.frame(x=data$x))
lines(transformedX
      ,predictions
      ,col='red')
```

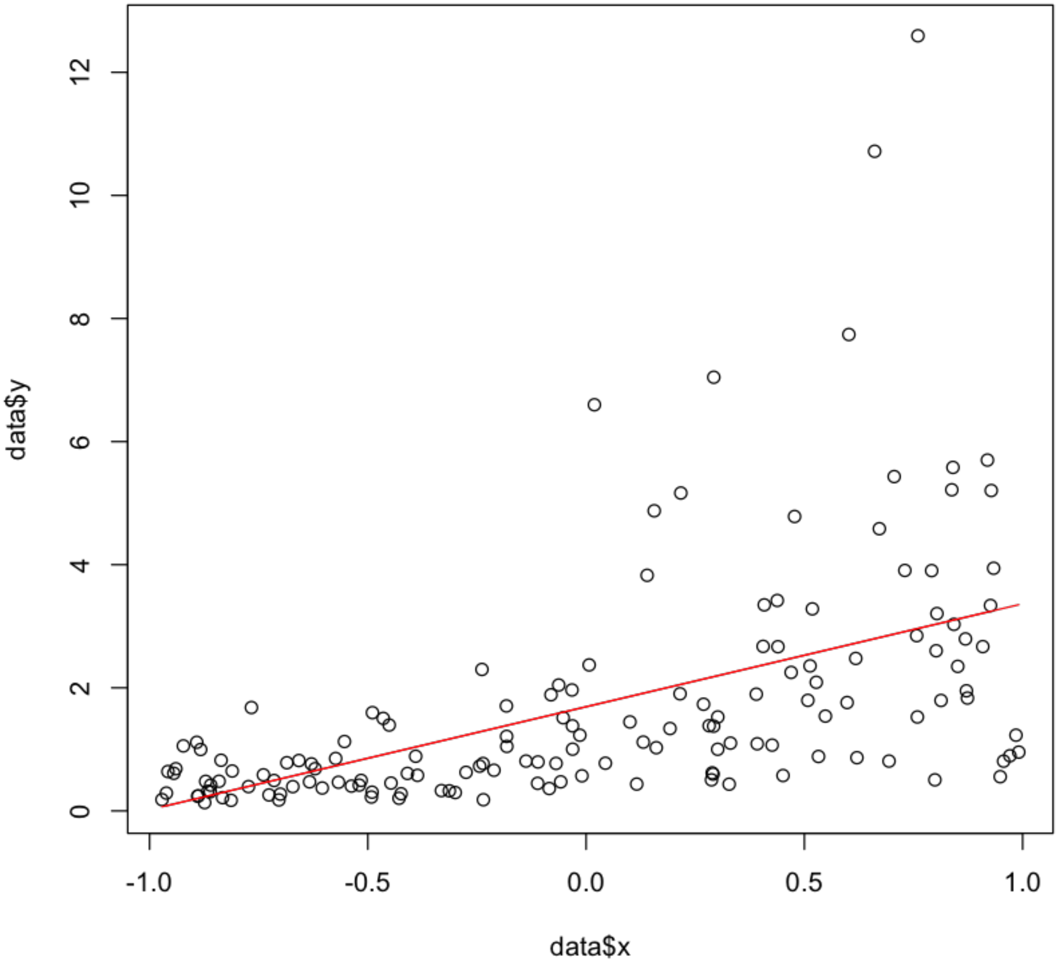
```
Warning message in log(data$x):
âĂIJNaNs producedâĂİ
Warning message in log(x):
âĂIJNaNs producedâĂİ
Warning message in log(x):
âĂIJNaNs producedâĂİ
```



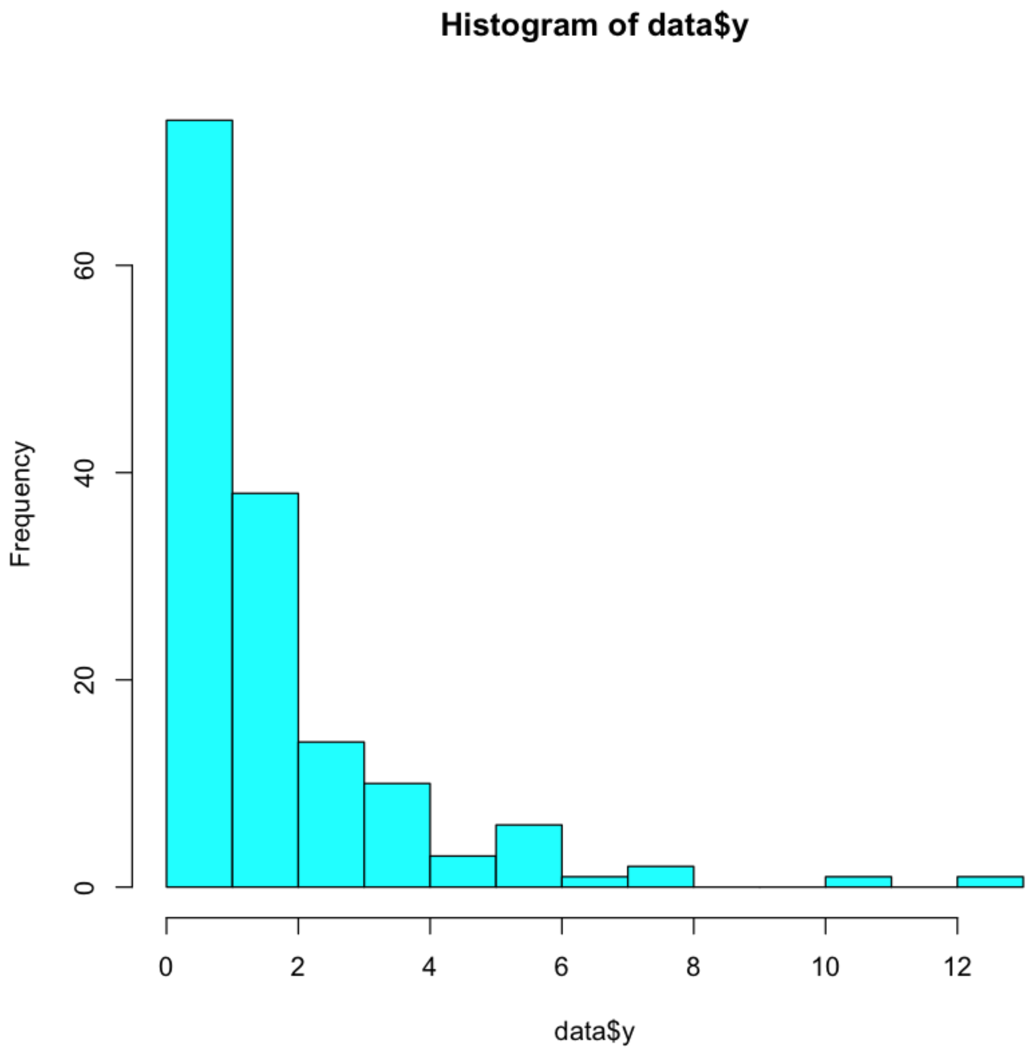
Common transformations to try applying to *x* are: log transform, exp transform, and x raised to some power.

0.4 Checking Normality (transforming y)

[71]:
```
data <- read.csv("./dataSet2.csv")
plot(data$x,data$y)
model1 <- lm(y~x,data=data)
lines(data$x,predict.lm(model1,data.frame(x=data$x)),col='red')
```



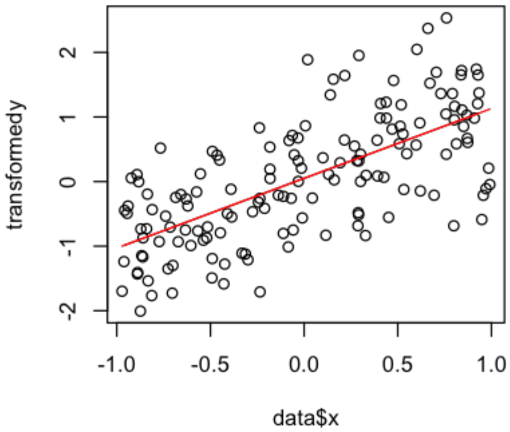
[72]:
```
hist(data$y,10,col='cyan')
```



[74]:

```
par(mfrow=c(3,2))

transformedy = log(data$y)
plot(data$x,transformedy)
model <- lm(transformedy~x,data=data)

predictions <- predict.lm(model,data.frame(x=data$x))
lines(data$x
      ,predictions
      ,col='red')

hist()
```

```
transformedy = log(data$y)
plot(data$x,transformedy)
model <- lm(transformedy~x,data=data)

predictions <- predict.lm(model,data.frame(x=data$x))
lines(data$x
      ,predictions
      ,col='red')

transformedy = log(data$y)
plot(data$x,transformedy)
model <- lm(transformedy~x,data=data)

predictions <- predict.lm(model,data.frame(x=data$x))
lines(data$x
      ,predictions
      ,col='red')
```



0.5 The QQ-plot

```
[ ]:
```