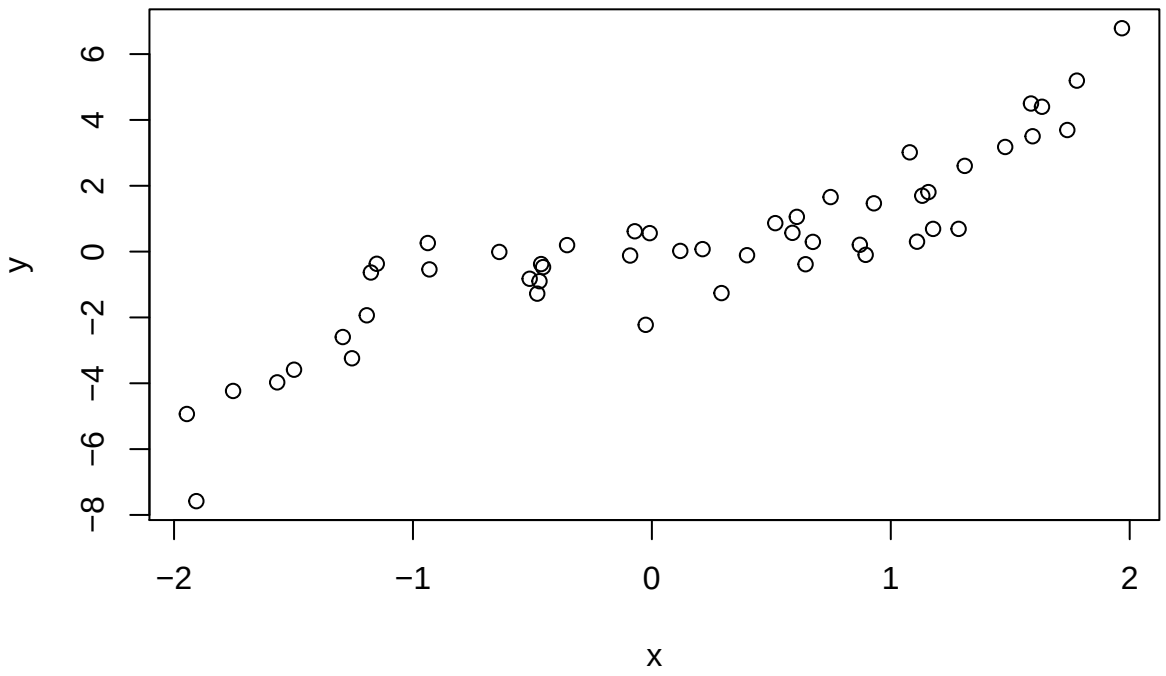
DaigleInClassLabWk10D1.R

2011home

Mon Mar 26 10:44:37 2018

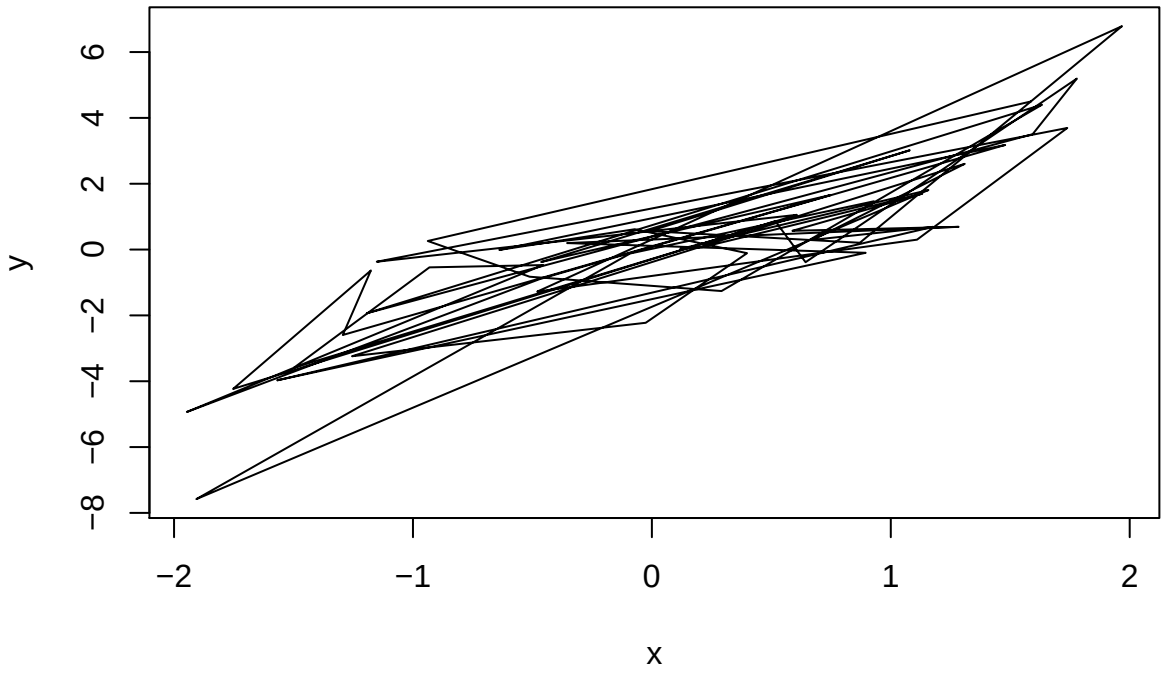
```
# Chris Daigle
# Week 10 Day 1, 26 March 2018
# In class lab

# Exercise 1 ####
# Explain: why does the plot() result with the type = "p" look normal, but the plot()
# result with type = "l" look abnormal.
# Modify the code below so that the lines on the second plot do not cross

n <- 50
set.seed(0)
x <- runif(n, min = -2, max = 2)
y <- x^3 + rnorm(n)

plot(x, y, type = "p")
```




```
graphics.off
```

```
## function ()
## {
##     while ((which <- dev.cur()) != 1) dev.off(which)
##     invisible()
## }
## <bytecode: 0x7f7fc5614590>
## <environment: namespace:grDevices>
```

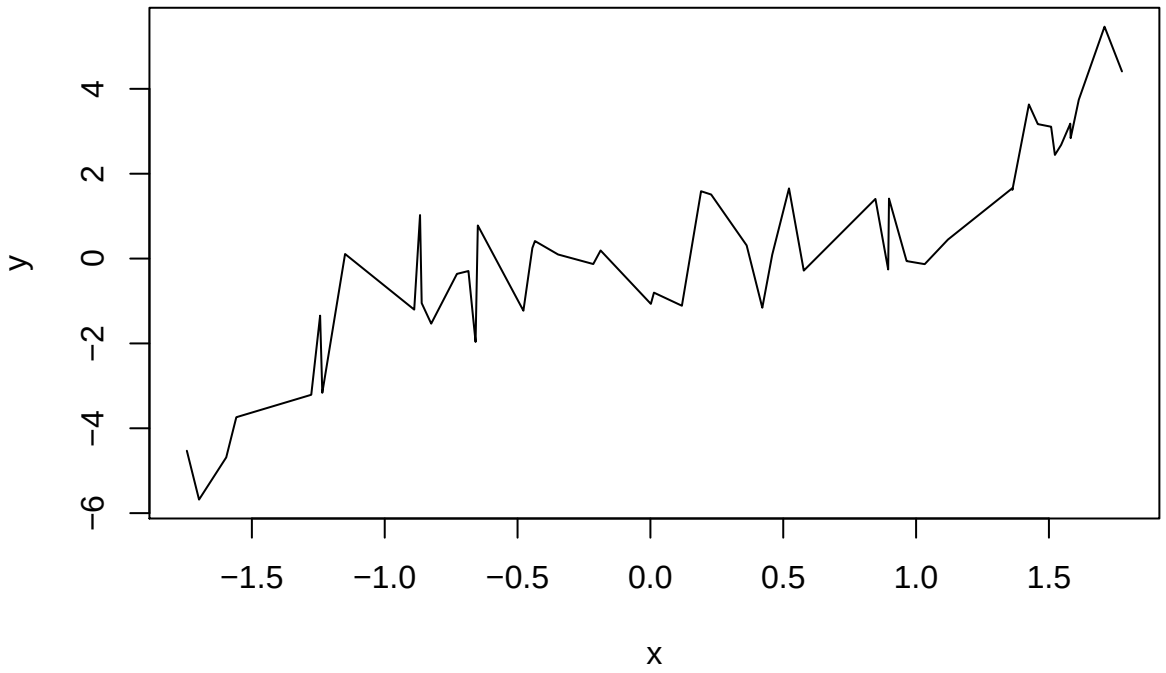
```r
plot(x, y, type = "l")
```




```r
graphics.off
```

```
## function ()
## {
##     while ((which <- dev.cur()) != 1) dev.off(which)
##     invisible()
## }
## <bytecode: 0x7f7fc5614590>
## <environment: namespace:grDevices>
```

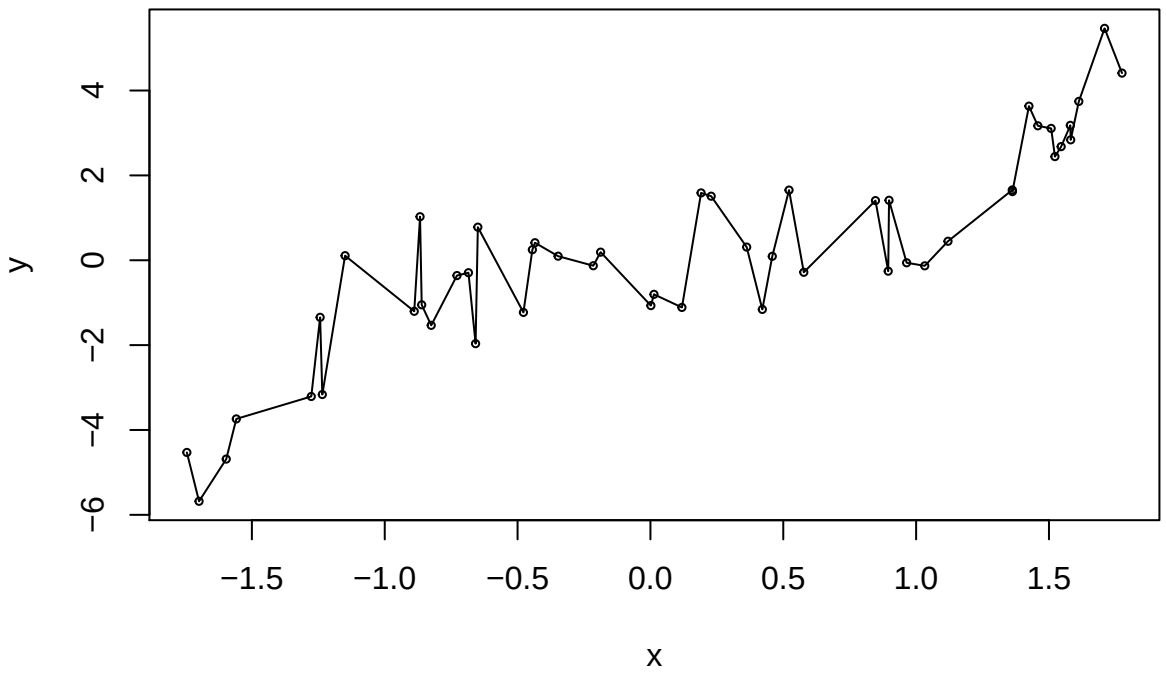
```r
# The reason this appears abnormal is that the line is connected to the points associated
# with x as they are generated, not in an ordinal nature. To fix this, we can order the
# input variable, x, by using sort() and then plot.

x <- sort(runif(n, min = -2, max = 2))
y <- x^3 + rnorm(n)
plot(x, y, type = "l")
```



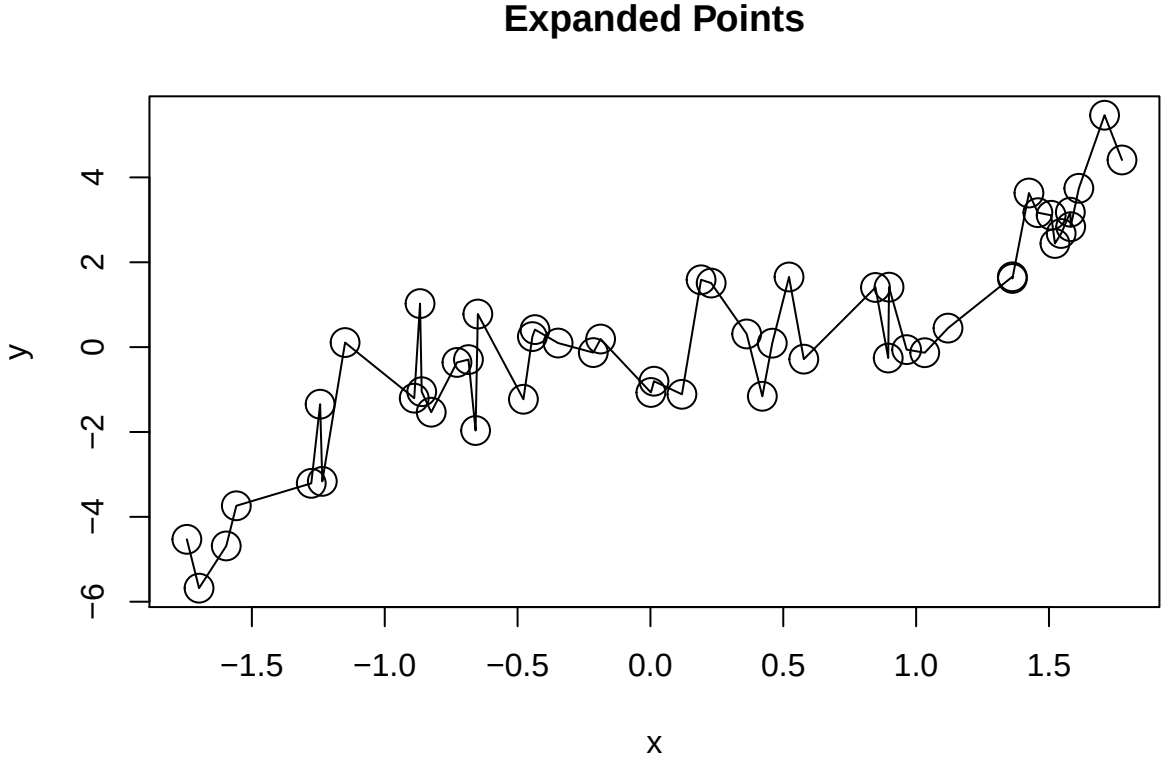

```
graphics.off
```

```
## function ()
## {
##     while ((which <- dev.cur()) != 1) dev.off(which)
##     invisible()
## }
## <bytecode: 0x7f7fc5614590>
## <environment: namespace:grDevices>
```

```
# The line now appears as expected.

# Exercise 2 ####
# Using cex. to scale values.
# 1: Plot y versus x with cex = 1/2 and label it "Shrunken Points"
# 2: Plot y versus x with cex = 2 and label it "Expanded Points"

plot(x, y, type = 'o', cex = 0.5, main = 'Shrunken Points')
```




```
plot(x, y, type = 'o', cex = 2, main = 'Expanded Points')
```

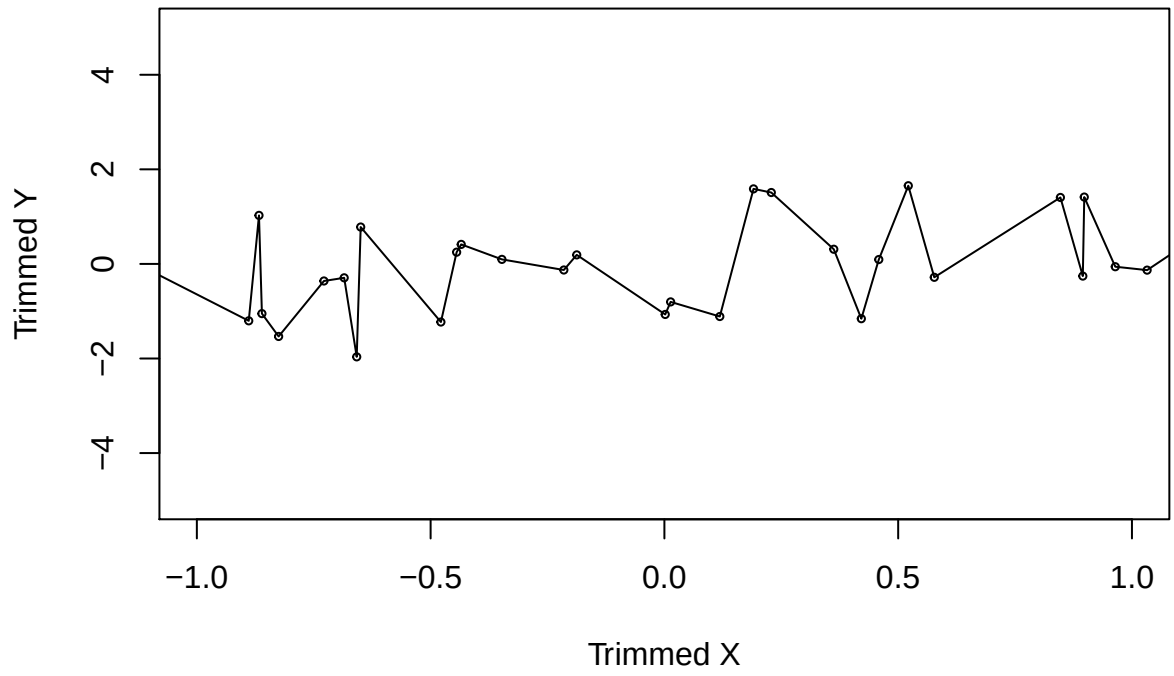



```
graphics.off
```

```
## function ()
## {
```

```
##     while ((which <- dev.cur()) != 1) dev.off(which)
##     invisible()
## }
## <bytecode: 0x7f7fc5614590>
## <environment: namespace:grDevices>
```

```
# Exercise 3 ####
# Using bounds, limits, on the plot
# 1: Plot y vs x and set the x limit to [-1,1] and the y limit to [-5,5]
# 2: Label the axes "Trimmed X" and "Trimmed Y" respectively

plot(x, y, type = 'o', xlim = c(-1,1), ylim = c(-5, 5), xlab = 'Trimmed X', ylab = 'Trimmed Y', cex = 0
```

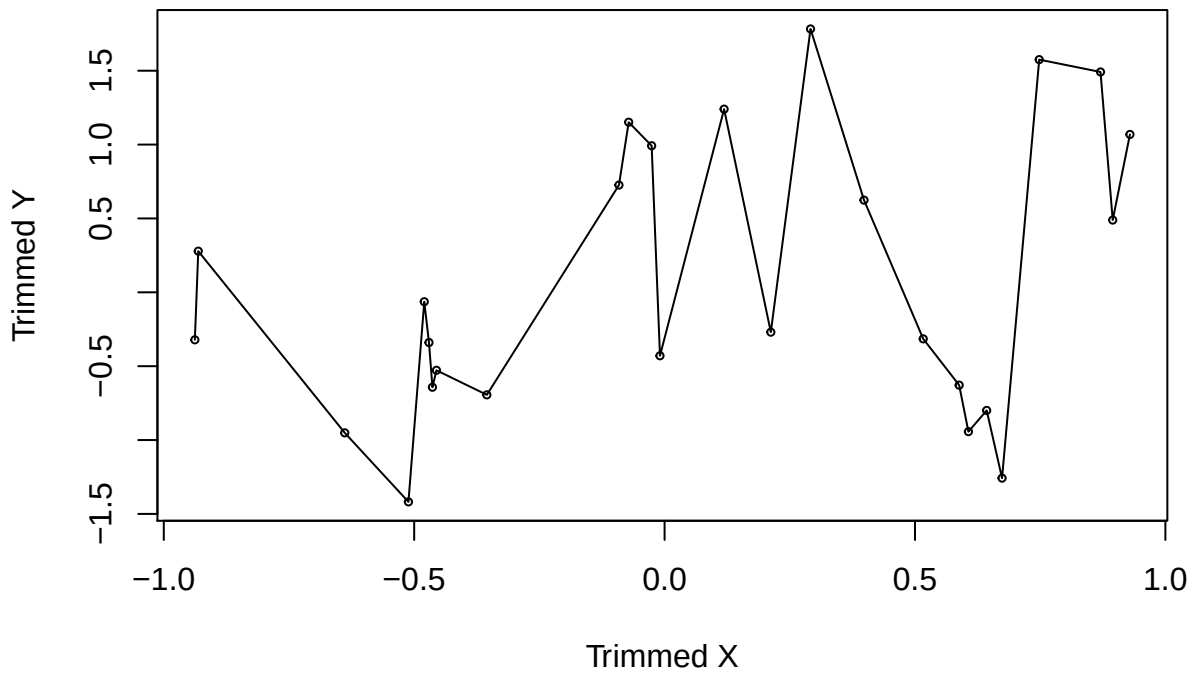



```
graphics.off
```

```
## function ()
## {
##     while ((which <- dev.cur()) != 1) dev.off(which)
##     invisible()
## }
## <bytecode: 0x7f7fc5614590>
## <environment: namespace:grDevices>
```

```
# Exercise 4 ####
# Define subsets of data to plot instead of using limits
# 1: Plot the same output as in Exercise 3, by subsetting x and y
# (x.trimmed <- ?, and y.trimmed <- ?) with a logical

# logic.x <- (x>= -1 & x <= 1)
# x.trimmed <- x * logic.x
# again <-x[which((x>= -1 & x <= 1))]
set.seed(0)
x <- sort(runif(n, min = -2, max = 2))
trimmed.x <- x[x> -1 & x < 1]
```

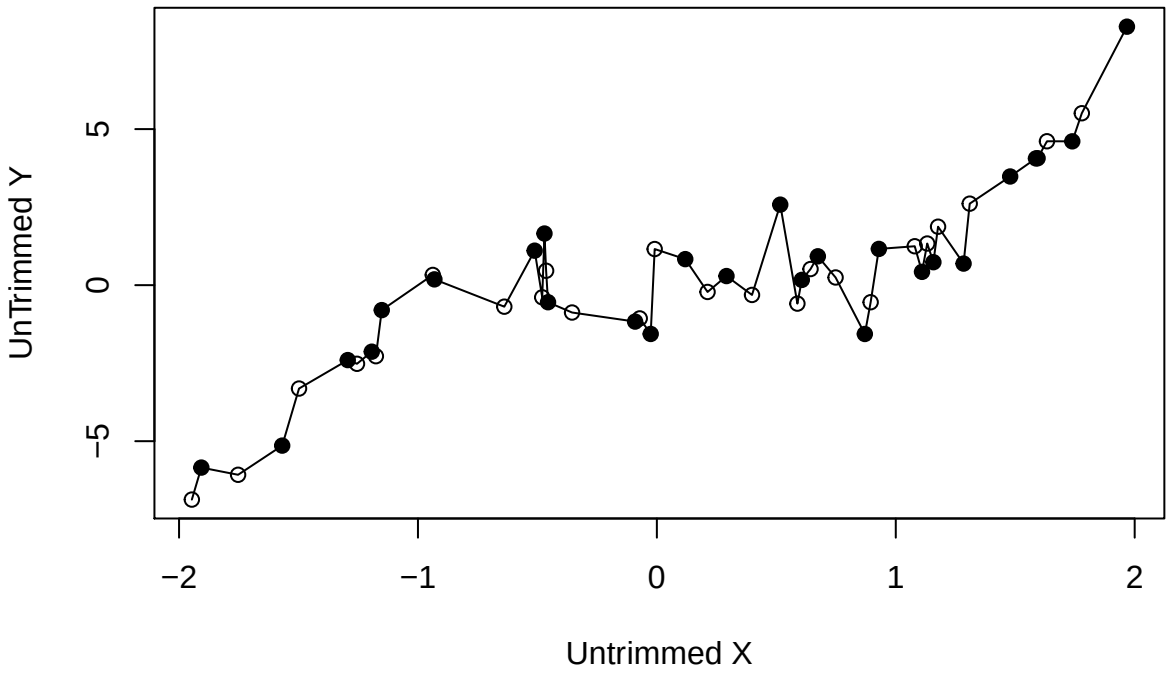
```r
trimmed.y <- trimmed.x^3 + rnorm(length(trimmed.x))

plot(trimmed.x, trimmed.y, type = 'o', cex = 1/2, xlab = 'Trimmed X', ylab = 'Trimmed Y')
```




```r
graphics.off
```

```
## function ()
## {
##     while ((which <- dev.cur()) != 1) dev.off(which)
##     invisible()
## }
## <bytecode: 0x7f7fc5614590>
## <environment: namespace:grDevices>
```

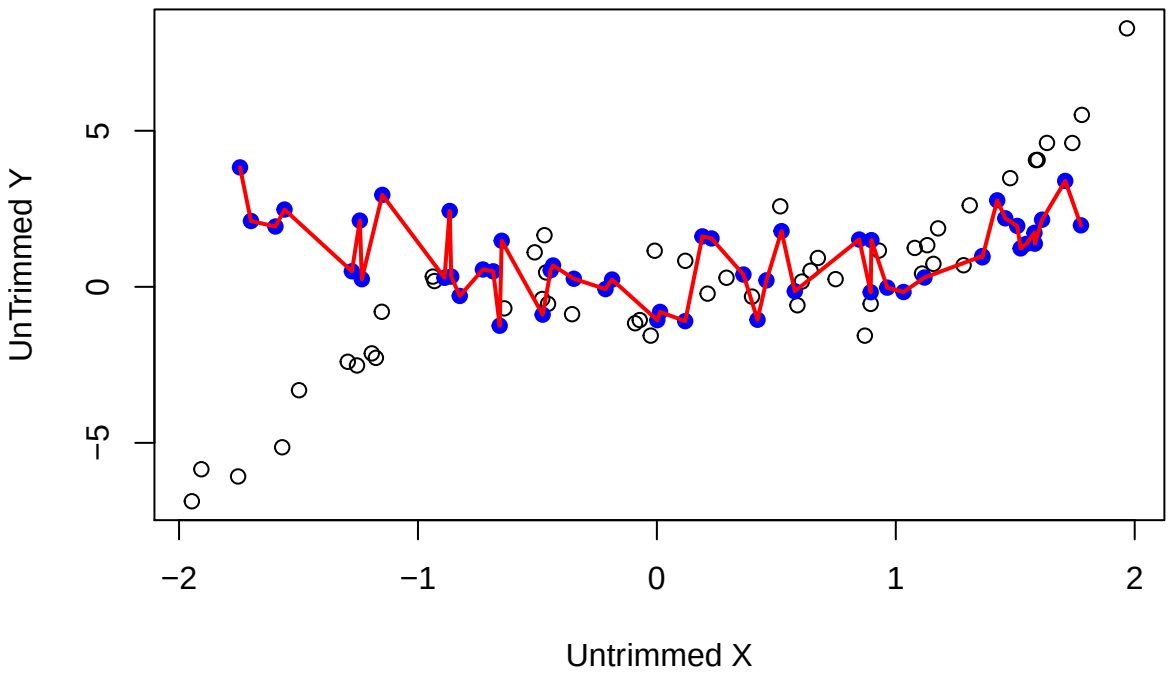
```r
# Exercise 5 ####
# Using pch

circleFillType <- c(1, 19)

n <- 50
set.seed(0)
x <- sort(runif(n, min = -2, max = 2))
y <- x^3 + rnorm(n)

plot(x, y, type = 'o', xlab = 'Untrimmed X', ylab = 'UnTrimmed Y', pch = circleFillType)
```




```
graphics.off
```

```
## function ()
## {
##     while ((which <- dev.cur()) != 1) dev.off(which)
##     invisible()
## }
## <bytecode: 0x7f7fc5614590>
## <environment: namespace:grDevices>
```

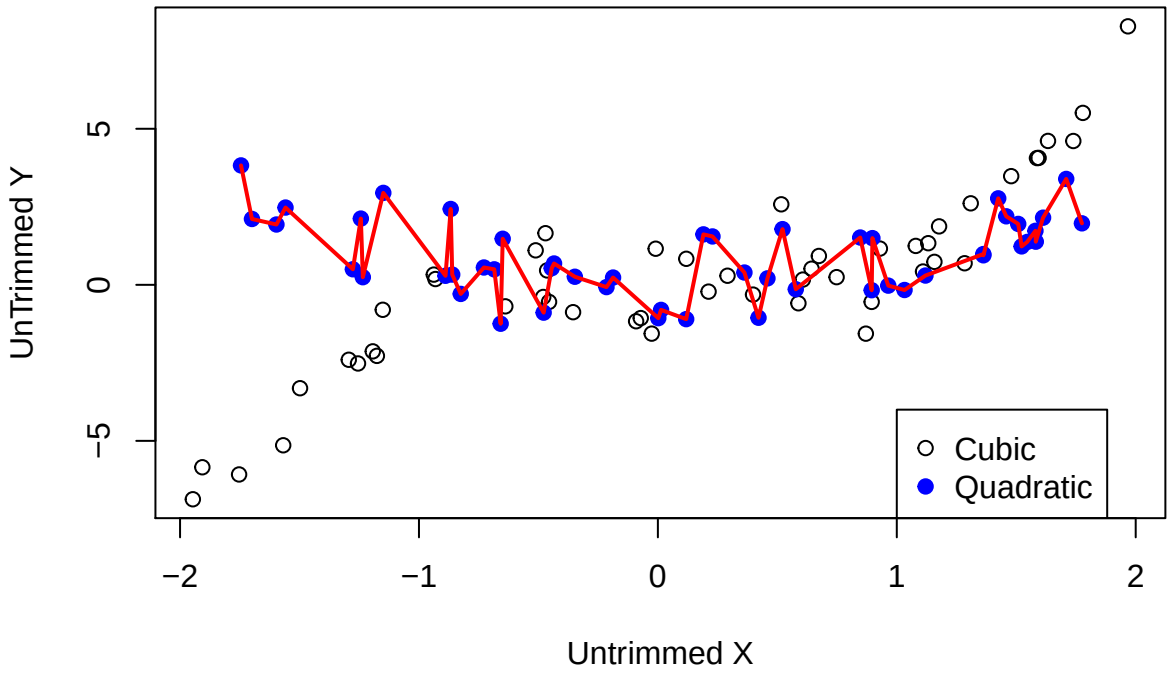
```
# Exercise 6 ####
# Adding to plots
# 1: Make a scatter of y vs x
# 2: Overlay x2 & y2
# 3: Set the x2 & y2 points to blue filled circles

n <- 50
set.seed(0)
x <- sort(runif(n, min = -2, max = 2))
y <- x^3 + rnorm(n)

x2 <- sort(runif(n, min = -2, max = 2))
y2 <- x2^2 + rnorm(n)

plot(x, y, xlab = 'Untrimmed X', ylab = 'UnTrimmed Y', main = "Overlaid X2 & Y2")
points(x2, y2, pch = 19, col = "blue")

# Exercise 7 ####
# 1: From the previous plot, overlat a line plot of y2 vs x2

lines(x2, y2, col = "red", lwd = 2)
```



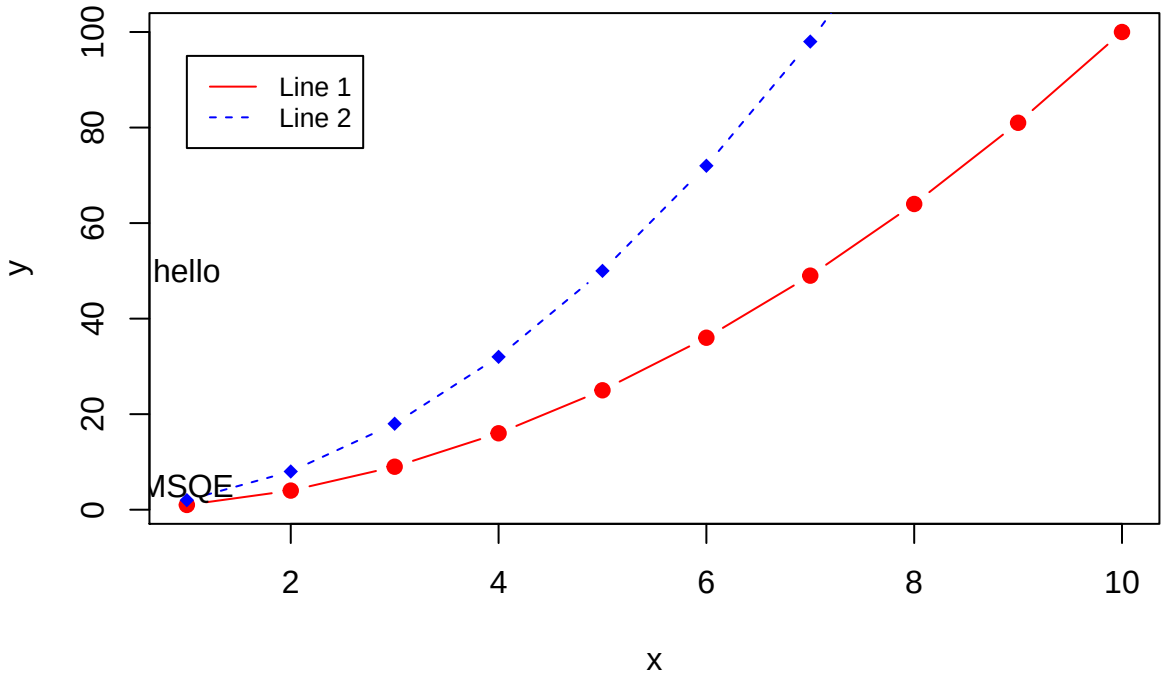

```
# Exercise 8 ####
# Adding a legend
# 1: Create a legend that displays "Cubic" with empty black circles and "Quadratic" with
# the filled blue circles
graphics.off
```

```
## function ()
## {
##     while ((which <- dev.cur()) != 1) dev.off(which)
##     invisible()
## }
## <bytecode: 0x7f7fc5614590>
## <environment: namespace:grDevices>
```

```
n <- 50
set.seed(0)
x <- sort(runif(n, min = -2, max = 2))
y <- x^3 + rnorm(n)

x2 <- sort(runif(n, min = -2, max = 2))
y2 <- x2^2 + rnorm(n)

plot(x, y, xlab = 'Untrimmed X', ylab = 'UnTrimmed Y', main = "Overlaid X2 & Y2")
points(x2, y2, pch = 19, col = "blue")
lines(x2, y2, col = "red", lwd = 2)
legend(1, -4, legend = c("Cubic", "Quadratic"), col = c("black", "blue"), pch = c(1, 19))
```




```
# Exercise 9 ####
# Reproduce the code

x <- 1:10
y1 <- x^2
y2 <- 2*y1

plot(x, y1, type = 'b', pch = 19, col = 'red', xlab = 'x', ylab = 'y')
lines(x,y2, pch = 18, col = 'blue', type = 'b', lty = 2)
legend(1, 95, legend = c("Line 1", "Line 2"), col = c("red", "blue"), lty = 1:2, cex = 0.8)
text(1,50, labels = "hello")
text(1,5, labels = "MSQE")
```


Line 1
Line 2
hello
MSQE
y
x
0
20
40
60
80
100
2
4
6
8
10