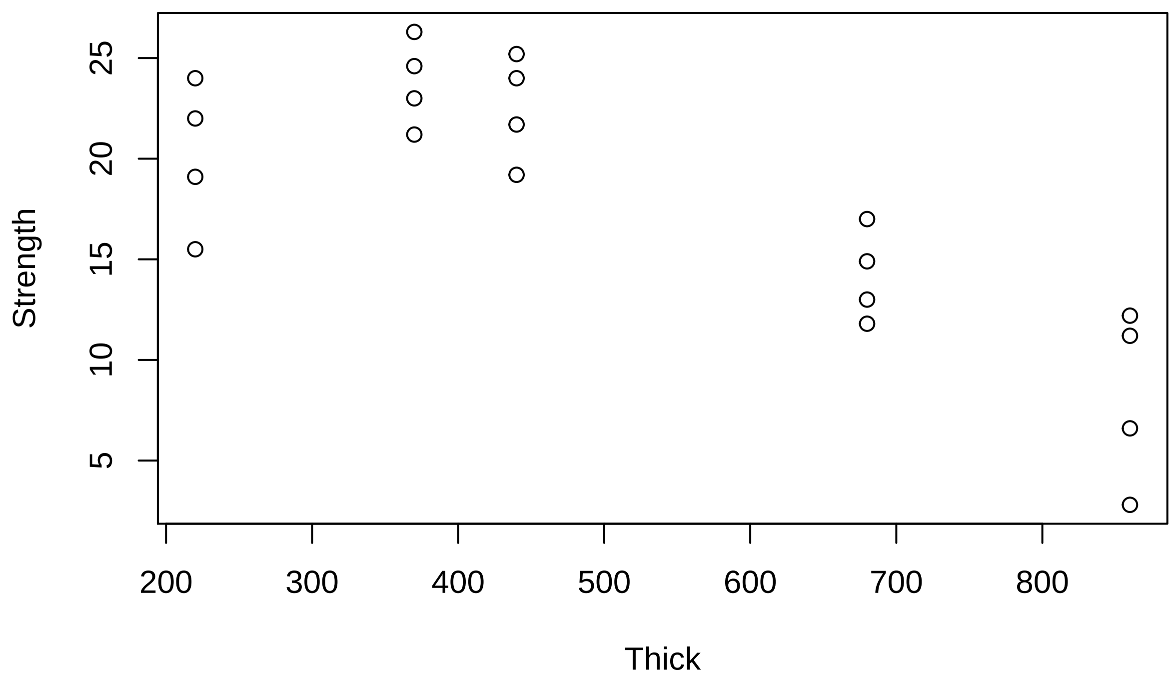
Steel Example: Quadratic Regression

An experiment was conducted to examine the relationship between Strength (Y) and coating Thickness (X) in steel. The scatterplot shows strong curvature, hence simple linear regression is not appropriate. Quadratic regression seems to fit the data well.

```
Steel <- read.csv("C:/hess/STAT512/RNotes/MultReg2/MR2_Steel.csv")
str(Steel)
```

```
## 'data.frame':    20 obs. of  2 variables:
##  $ Thick   : int  220 220 220 220 370 370 370 370 440 440 ...
##  $ Strength: num  24 22 19.1 15.5 26.3 24.6 23 21.2 25.2 24 ...
```

```
plot(Strength ~ Thick, data = Steel)
```



Linear Regression

```
Model1 <- lm(Strength ~ Thick, data = Steel)
summary(Model1)
```

```
##
## Call:
## lm(formula = Strength ~ Thick, data = Steel)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -8.8530 -2.2722  0.5315  2.4463  5.7768
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 29.282737   2.258464  12.966 1.44e-10 ***
## Thick       -0.022408   0.004016  -5.579 2.70e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.095 on 18 degrees of freedom
## Multiple R-squared:  0.6336, Adjusted R-squared:  0.6132
## F-statistic: 31.13 on 1 and 18 DF,  p-value: 2.699e-05
```

Quadratic Regression

The I() operator tells R to use the result of the calculaton, not the formula. Another option is to use the poly() function. Not shown here.

```r
Model2 <- lm(Strength ~ Thick + I(Thick^2), data = Steel)
summary(Model2)
```

```
##
## Call:
## lm(formula = Strength ~ Thick + I(Thick^2), data = Steel)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6222 -2.1960  0.2443  2.4491  4.8763
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.452e+01  4.752e+00   3.057  0.00713 **
## Thick        4.318e-02  1.980e-02   2.181  0.04354 *
## I(Thick^2)  -5.994e-05  1.786e-05  -3.357  0.00374 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.268 on 17 degrees of freedom
## Multiple R-squared:  0.7796, Adjusted R-squared:  0.7537
## F-statistic: 30.07 on 2 and 17 DF,  p-value: 2.609e-06
```

```r
model.matrix(Model2)
```

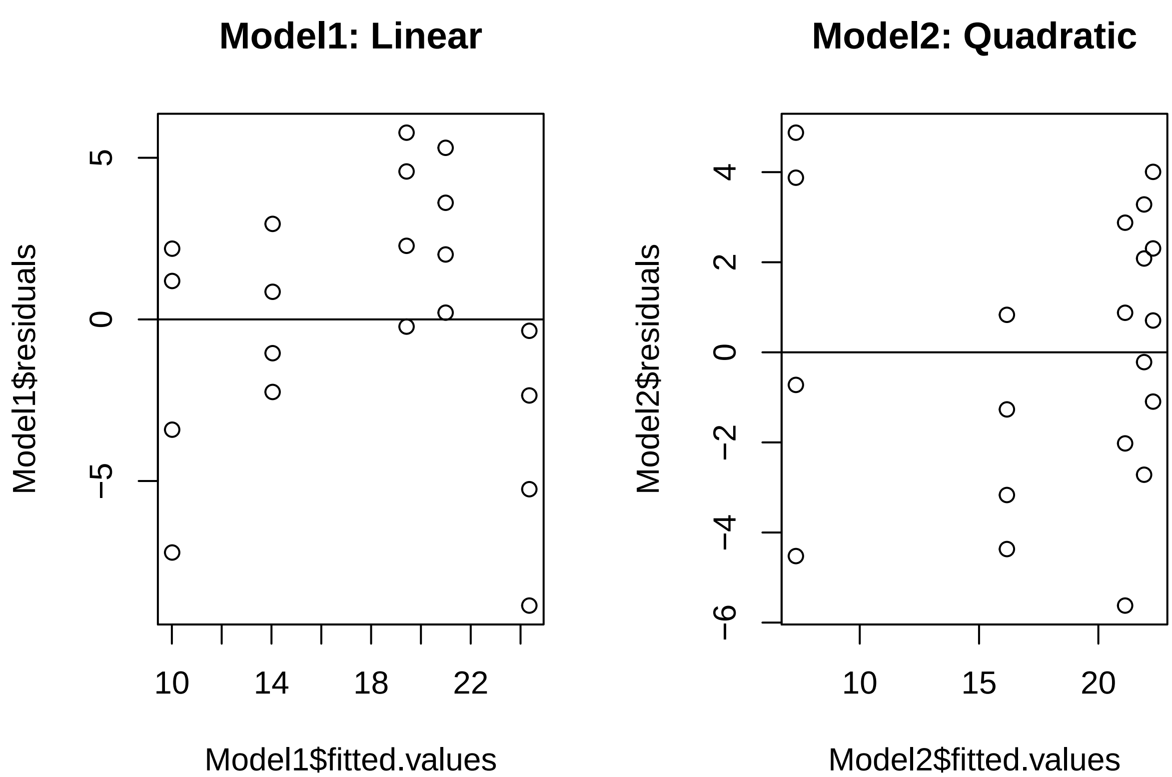
```
##    (Intercept) Thick I(Thick^2)
## 1            1   220      48400
## 2            1   220      48400
## 3            1   220      48400
## 4            1   220      48400
## 5            1   370     136900
## 6            1   370     136900
## 7            1   370     136900
## 8            1   370     136900
## 9            1   440     193600
## 10           1   440     193600
```

```
## 11            1    440      193600
## 12            1    440      193600
## 13            1    680      462400
## 14            1    680      462400
## 15            1    680      462400
## 16            1    680      462400
## 17            1    860      739600
## 18            1    860      739600
## 19            1    860      739600
## 20            1    860      739600
## attr(,"assign")
## [1] 0 1 2
```

Diagnostic plots

Resids vs Fitted values for both the linear and quadratic regressions. Note that I could also have used the plot() command directly (ex: plot(Model1)).

```
par(mfrow = c(1, 2))
plot(Model1$residuals ~ Model1$fitted.values)
abline(h=0)
title("Model1: Linear")
plot(Model2$residuals ~ Model2$fitted.values)
abline(h=0)
title("Model2: Quadratic")
```



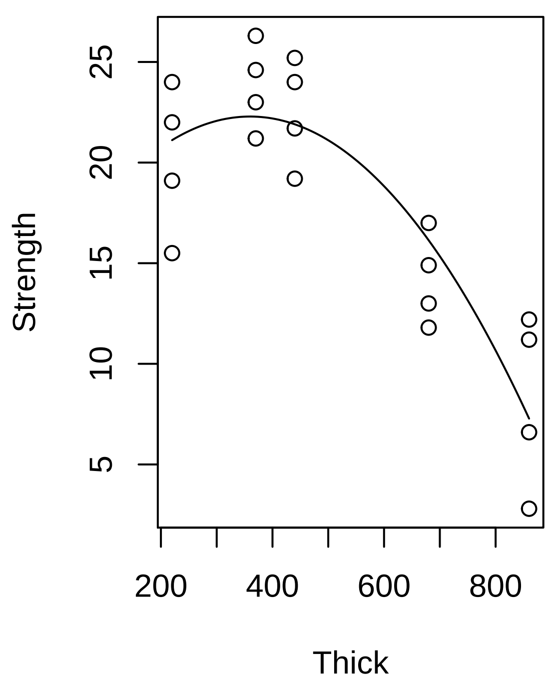
Overlaying the Fitted Curve

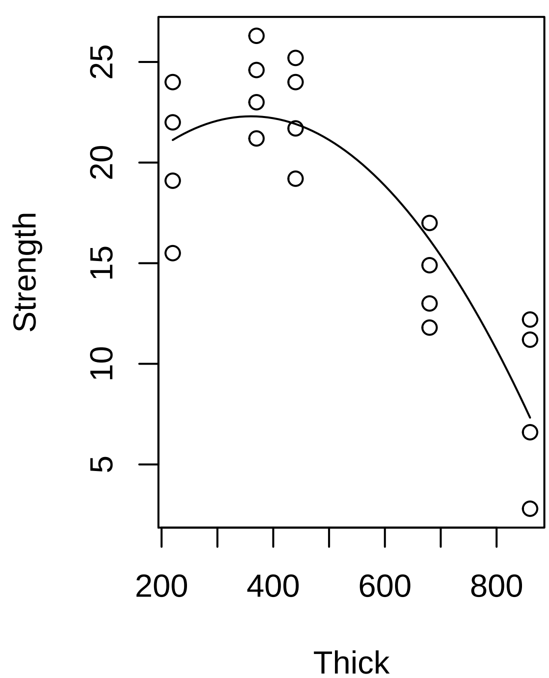
We illustrate three different approaches to overlaying the fitted curve.

```
#Approach 1: Using curve()
par(mfrow=c(1, 2))
plot(Strength ~ Thick, data = Steel)
curve(14.52 + 0.04318*x - 0.00006*x^2, add = TRUE)
#Approach 2: Using predict()
summary(Steel$Thick)
```

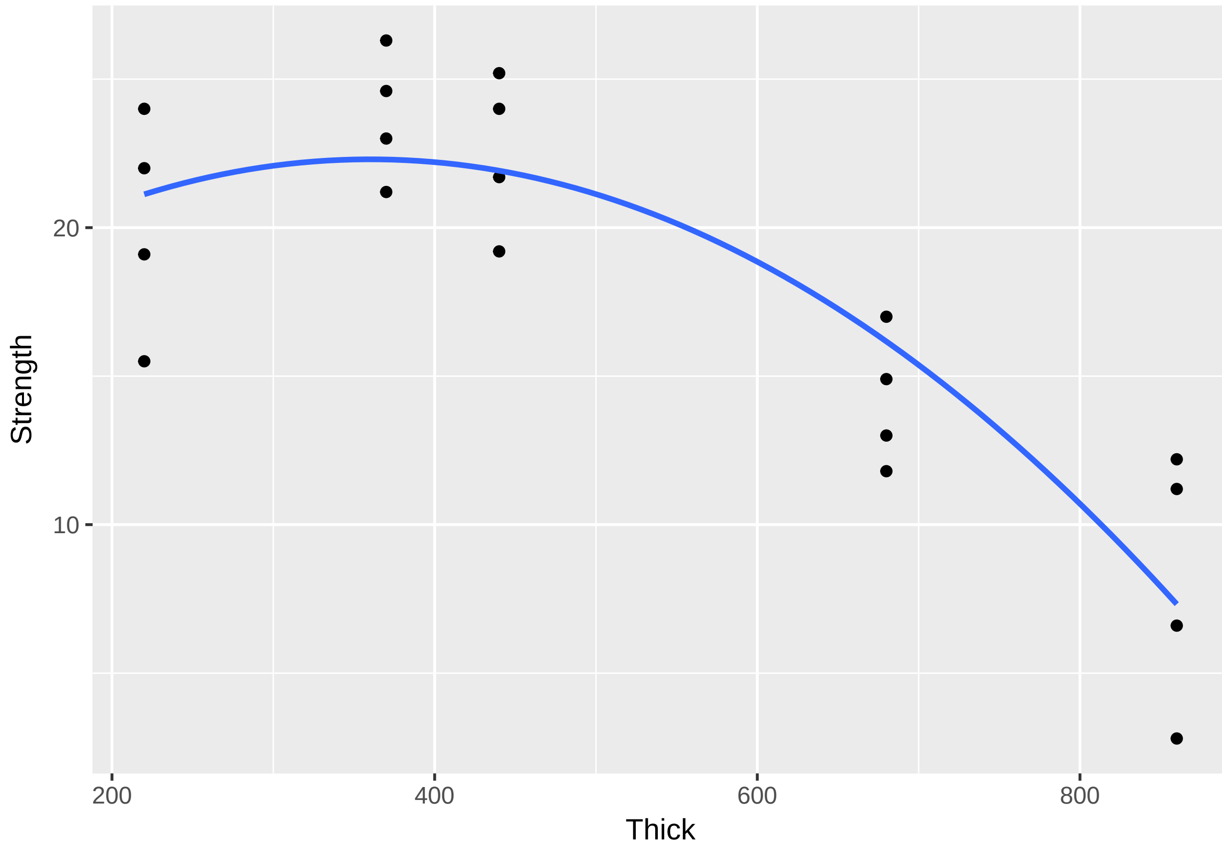
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     220     370     440     514     680     860
```

```
Xnew <- seq(from = 220, to = 860, by = 10)
Yhat <- predict(Model2, list(Thick = Xnew))
plot(Strength ~ Thick, data = Steel)
lines(Yhat ~ Xnew)
#Approach 3: Using ggplot2
library(ggplot2)
```






```
qplot(Thick, Strength, data = Steel) +
geom_smooth(method = "lm", formula = y ~ poly(x, 2), se = FALSE)
```


20
10
Strength
200
400
600
800
Thick