Regression with polynomial functions

For certain data sets, low-degree polynomial functions provide much better approximations than straight lines. Suppose we want to fit a polynomial to predict y from a single explanatory x. The model can be written as:

$$\hat{y} = b_0 + b_1 x + b_2 x^2 + \cdots + b_k x^k$$

Notice how easy it is to "pretend" that this is just a linear regression function in k predictors -- which we already know how to solve!

Example:

This example and data file are from the open web archives of the

[Statistics Department at Penn State University (https://online.stat.psu.edu/stat501/)](https://online.stat.psu.edu/stat501/)

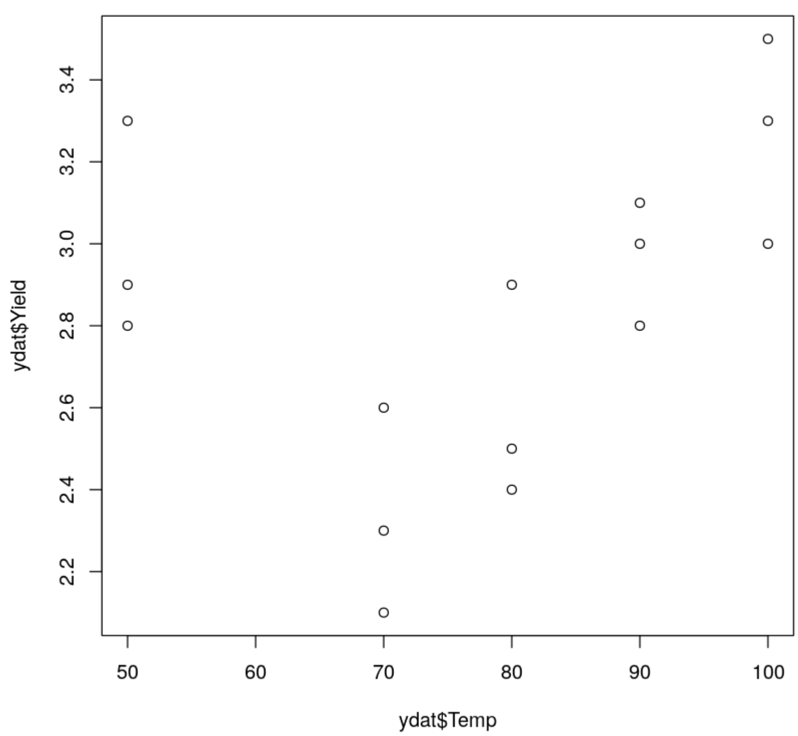
In this example, the data consists of measurements of crop yield from an experiment done at different temperatures. The variables are clearly labeled in the header of the data file (temperature is in $^\circ F$).

In [11]:
```
# Read/load data
ydat = read.csv(file="https://cs.earlham.edu/~pardhan/sage_and_r/yield.csv", header=TRUE, sep=",")
head (ydat)

# Explore shape via scatterplot.
plot(ydat$Yield ~ ydat$Temp)
```

A data.frame: 6 × 3

	i	Temp	Yield
	<int>	<int>	<dbl>
1	1	50	3.3
2	2	50	2.8
3	3	50	2.9
4	4	70	2.3
5	5	70	2.6
6	6	70	2.1



In [12]:
```
# Try to fit a linear model and see:
#
lmod = lm(Yield ~ Temp, data=ydat)
summary (lmod)
```

```
Call:
lm(formula = Yield ~ Temp, data = ydat)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.67928 -0.26306  0.05315  0.22072  0.65586 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.306306   0.469075   4.917 0.000282 ***
Temp        0.006757   0.005873   1.151 0.270641    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3913 on 13 degrees of freedom
Multiple R-squared:  0.09242,	Adjusted R-squared:  0.0226 
F-statistic: 1.324 on 1 and 13 DF,  p-value: 0.2706
```

Exercise: Discuss the effectiveness of this model by examining the usual evidence: the conditions; R^2; significance of various relevant results, etc.

In [20]:
```
# Now, let's try a quadratic fit: y = b0 + b1 x + b2 x^2
#
x1 = ydat$Temp
x2 = x1*x1
qmod = lm(ydat$Yield ~ x1+x2)
summary (qmod)
```

```
Call:
lm(formula = ydat$Yield ~ x1 + x2)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.37113 -0.15567 -0.04536  0.15790  0.35258 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  7.9604811  1.2589183   6.323 3.81e-05 ***
x1          -0.1537113  0.0349408  -4.399 0.000867 ***
x2           0.0010756  0.0002329   4.618 0.000592 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2444 on 12 degrees of freedom
Multiple R-squared:  0.6732,	Adjusted R-squared:  0.6187 
F-statistic: 12.36 on 2 and 12 DF,  p-value: 0.001218
```

Exercise: Discuss the effectiveness of this model -- look at R^2; significance of slopes, etc.

Great! Now, let's try a cubic and see if things get even better!!!

In [ ]:
```

```

In [ ]:
```

```