Polynomial regression

Instead of transforming responses, another possibility is to either transform predictor variables or add polynomial functions of predictor variables.

- This is especially done when the relationship between response and predictors appears to be curvilinear
- In this case the response might be considered a polynomial (e.g., quadratic, cubic, etc.) function of the predictor(s).
- We can fit a quadratic, cubic, etc. relationship by defining squares, cubes, etc., and use them as additional explanatory variables
- We can also do this with more than one explanatory variable, in which case we also often include an interaction term. When we do this we generally create a multicollinearity problem, which can often be corrected by standardization or centering.

Figure: Simulated data study

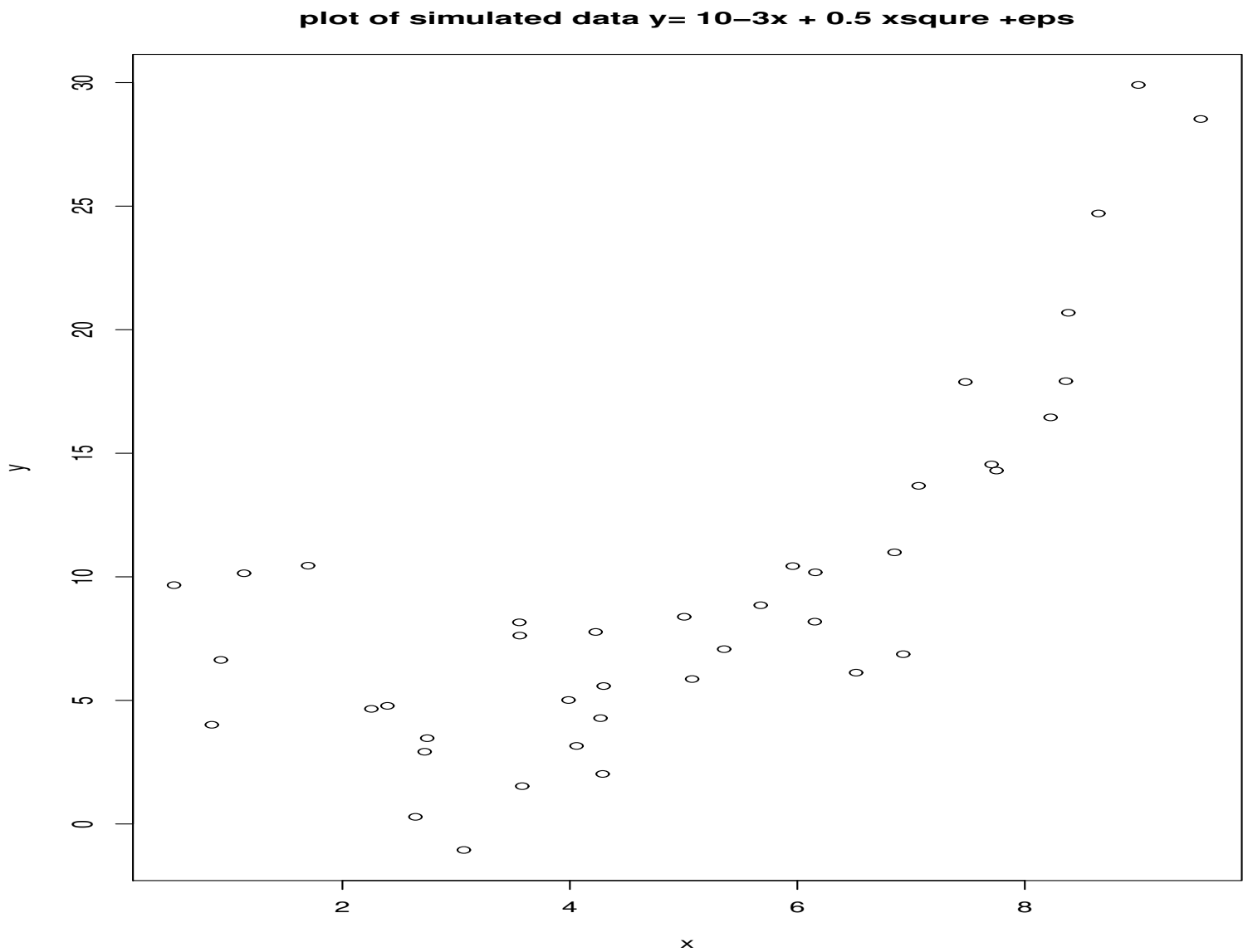
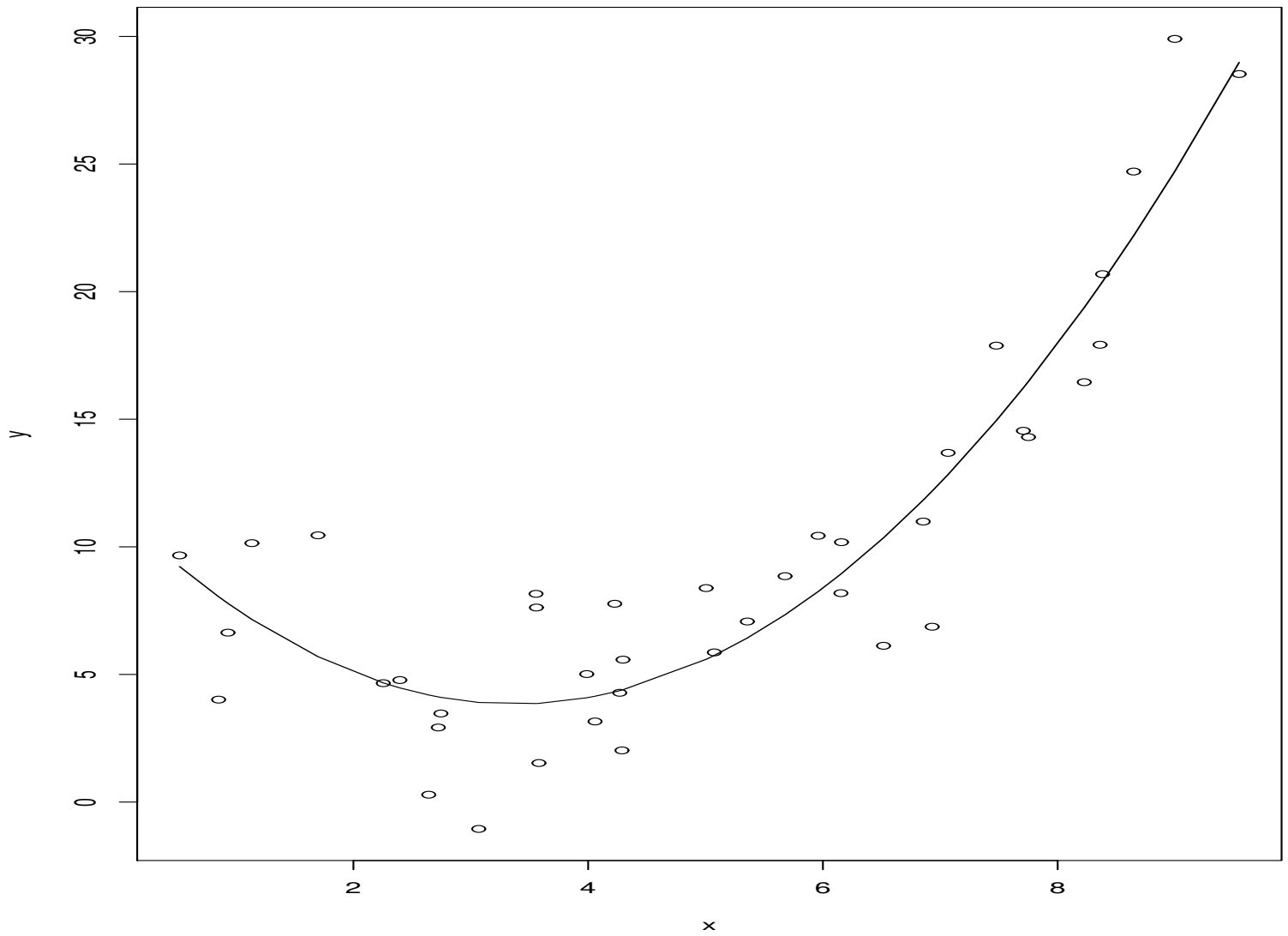


Figure: Simulated data study 2



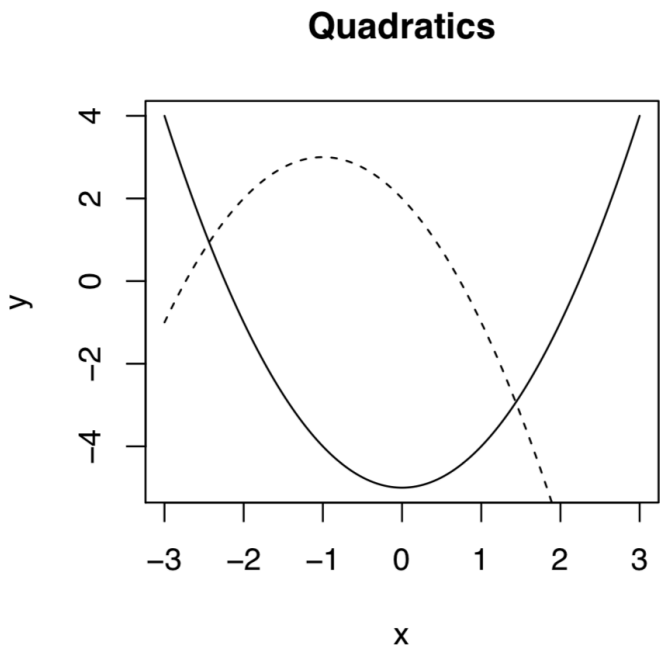
Polynomial regression

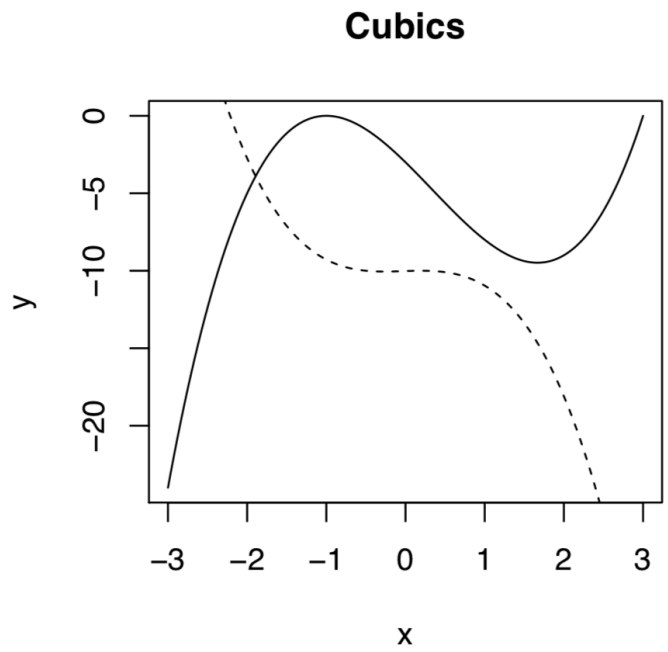
If there is only one predictor variable, the model can be written as

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_p x_i^p + \varepsilon$$

Often, only $p = 2$ or $p = 3$ is used. Here are some examples in R:

```
x <- seq(-3,3,0.01);
y21 <- x^2-5;
y22 <- -(x+1)^2+3;
y31 <- (x+1)^2*(x-3);
y32 <- -(x-.2)^2*(x+.5)-10;
plot(  x, y21, type="l", main="Quadratics", ylab="y")
points(x, y22, type="l", lt=2)
plot(  x, y31, type="l", main="Cubics", ylab="y")
points(x, y32, type="l", lt=2)
```


Quadratics
y
4
2
0
−2
−4
−3
−2
−1
0
1
2
3
x


Cubics
y
0
−5
−10
−20
−3
−2
−1
0
1
2
3
x

Polynomial regression

Note that

- a polynomial relationship might be useful even if the maximum and minimum points are not within the range of the predictor (for example on the left hand graph if only $x > 0$ is observed),
——- simply because it allows a nonlinear relationship.
- A linear regression is a special case of polynomial regression. For example, in the model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$$

——-If $\beta_2 = 0$, then the model reduces to simple linear regression.
—— Testing the null hypothesis $H_0 : \beta_2 = 0$ in this case could be used to decide whether the relationship is linear versus quadratic.

Polynomial regression

Polynomial regression can be fit by using new predictor variables based on powers of the original predictor `x`. Here is a toy example using powers up to x^4 with only 5 observations:

```
> x <- rnorm(5)
> y <- x+runif(5)
> x2 <- x^2
> x3 <- x^3
> x4 <- x^4
> myfit <- lm(y ~ x + x2 + x3 + x4)
> summary(a)
> x
[1]  0.6292986  0.6346305 -0.2228644 -1.1363222 -0.8370428
> y
[1] -0.05138843  2.54694578  0.91717318 -1.25256085 -0.66412
```

```
> summary(myfit)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -28.190         NA      NA       NA
x           -145.540         NA      NA       NA
x2            -1.357         NA      NA       NA
x3           343.781         NA      NA       NA
x4           220.553         NA      NA       NA

Residual standard error: NaN on 0 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:    NaN
F-statistic:   NaN on 4 and 0 DF,  p-value: NA
```

Polynomial regression

- with n observations, a polynomial regression with $p = n - 1$ predictors can **exactly** fit the data.
  ——it tends to make some extreme predictions for potential data values that weren't observed.
- it doesn't allow any extra information to estimate uncertainty. As a result, the standard errors and p-values cannot be given.
- This results in overfitting. The idea of overfitting is that the model fits the particular observations but is unlikely to generalize to a new data set collected from the same population.
- extrapolation beyond the range of the data can be dangerous in linear regression, the situation is even worse in polynomial regression since it can lead to extreme predictions.
- Another issue in polynomial regression is that measurement scale (e.g., Celsius versus Fahrenheit), now can affect the results (p-vlaues, predicted values, etc.).

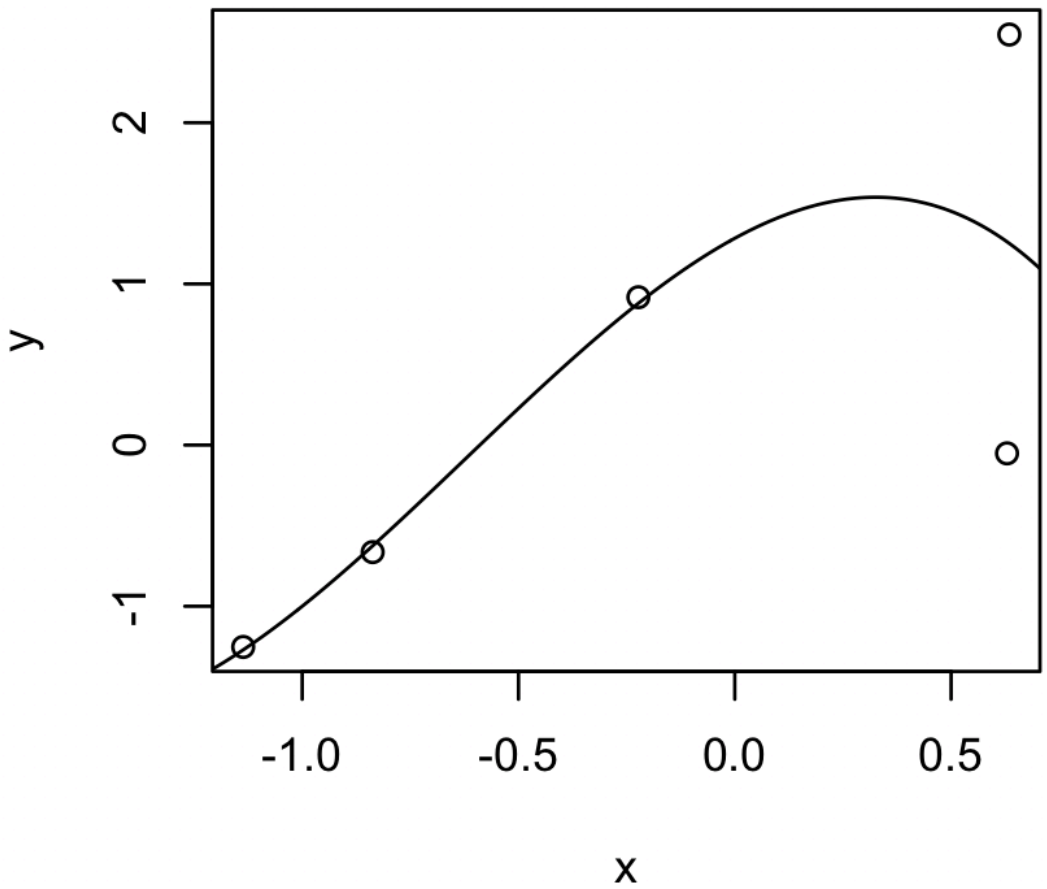
Polynomial regression

Amazingly, by having just one less parameter, you can suddenly get standard errors and p-values for all parameters.

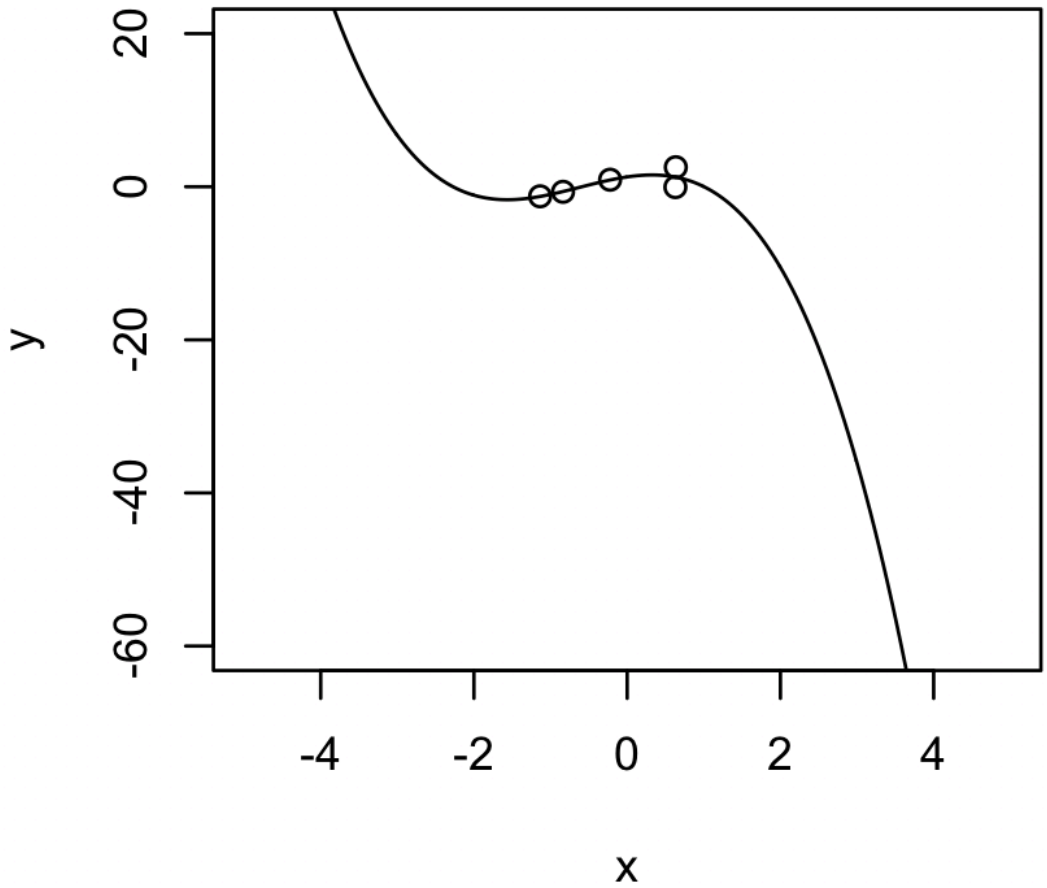
- none of the p-values indicates significance even though the curve essentially goes through three of the five data points. With a small ratio of sample size to parameters, it is difficult to find significance.

```
> a2 <- lm(y ~ x + x2 + x3)
> summary(a2)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.2830     2.5999   0.493    0.708
x             1.4618     4.7427   0.308    0.810
x2           -1.7765     7.4583  -0.238    0.851
x3           -0.9551     8.7390  -0.109    0.931
```

Polynomial regression: cubic fit



Polynomial regression: cubic fit



Polynomial regression: cubic fit

We also see that interpolation seems more reasonable in the cubic model compared to the quartic, but that extrapolation (beyond the range of the data) will lead to some very extreme predictions.

Polynomial regression: cubic fit

Although a simple linear regression is also not significant, the p-values have gone down and the standard errors are much smaller.

```
> a3 <- lm(y ~ x)
> summary(a3)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.5536     0.5283   1.048    0.372
x             1.3643     0.7009   1.946    0.147

Residual standard error: 1.145 on 3 degrees of freedom
Multiple R-squared:  0.5581,Adjusted R-squared:  0.4108
F-statistic: 3.788 on 1 and 3 DF,  p-value: 0.1468
```

It is also possible to have two or more predictors

- Each of which could be fit with quadratic, cubic or higher order terms.
- With more predictors, you could easily end up with huge numbers of parameters to estimate, which will require more data.
  —— Usually we want as few parameters as possible, and for polynomial regression, we usually want to just use quadratic or maybe cubic powers if possible.

With two predictors, each of which could be quadratic, we can have the model

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i}^2 + \beta_4 x_{2i}^2 + \beta_4 x_{1i} x_{2i} + \varepsilon_i$$

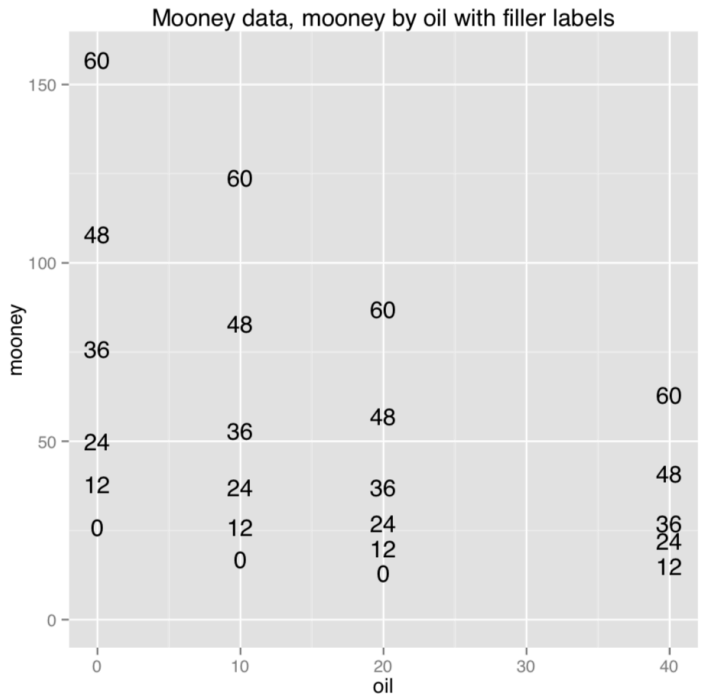
This model includes an interaction, which is still quadratic since the total power of $x_1 x_2$ is 2.

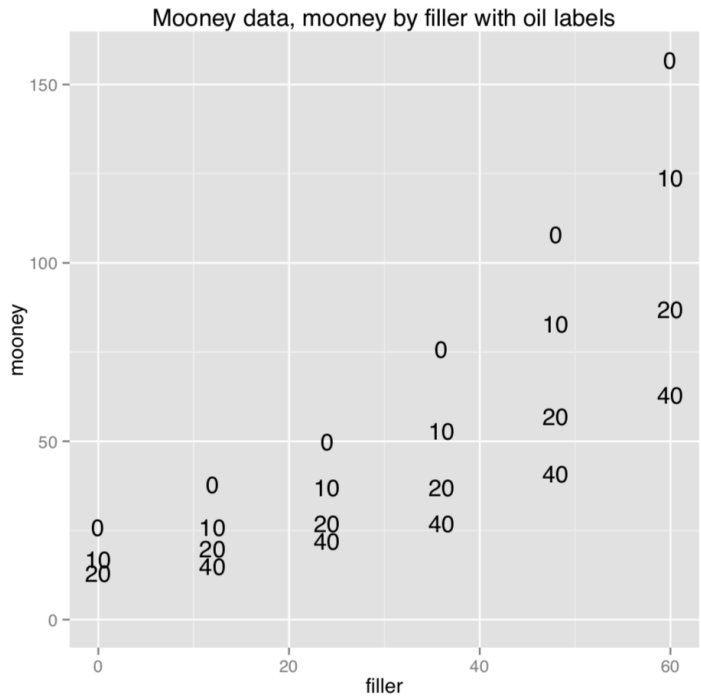
Mooney viscosity example

As an example, the data below give the Mooney viscosity at 100 degrees Celsius (y) as a function of the filler level (x_1) and the naphthenic oil (x_2) level for an experiment involving filled and plasticized elastomer compounds.

```
> mooney[1:10,]
   oil filler mooney
1    0      0     26
2    0     12     38
3    0     24     50
4    0     36     76
5    0     48    108
6    0     60    157
7   10      0     17
8   10     12     26
9   10     24     37
10  10     36     53
```

Polynomial regression





- At each of the 4 oil levels, the relationship between the Mooney viscosity and filler level (with 6 levels) appears to be quadratic.
- Similarly, the relationship between the Mooney viscosity and oil level appears quadratic for each filler level (with 4 levels).
- This supports fitting the general quadratic model as a first step in the analysis.

The graphic plots two variables—such as Mooney viscocity against oil, and instead of using a plotting character for each point, replaces it with the value of the third variable. This is a clever way to get three dimensional information into an apparently two-dimensional graph, and mostly works if you have a small number of values in the third variable.

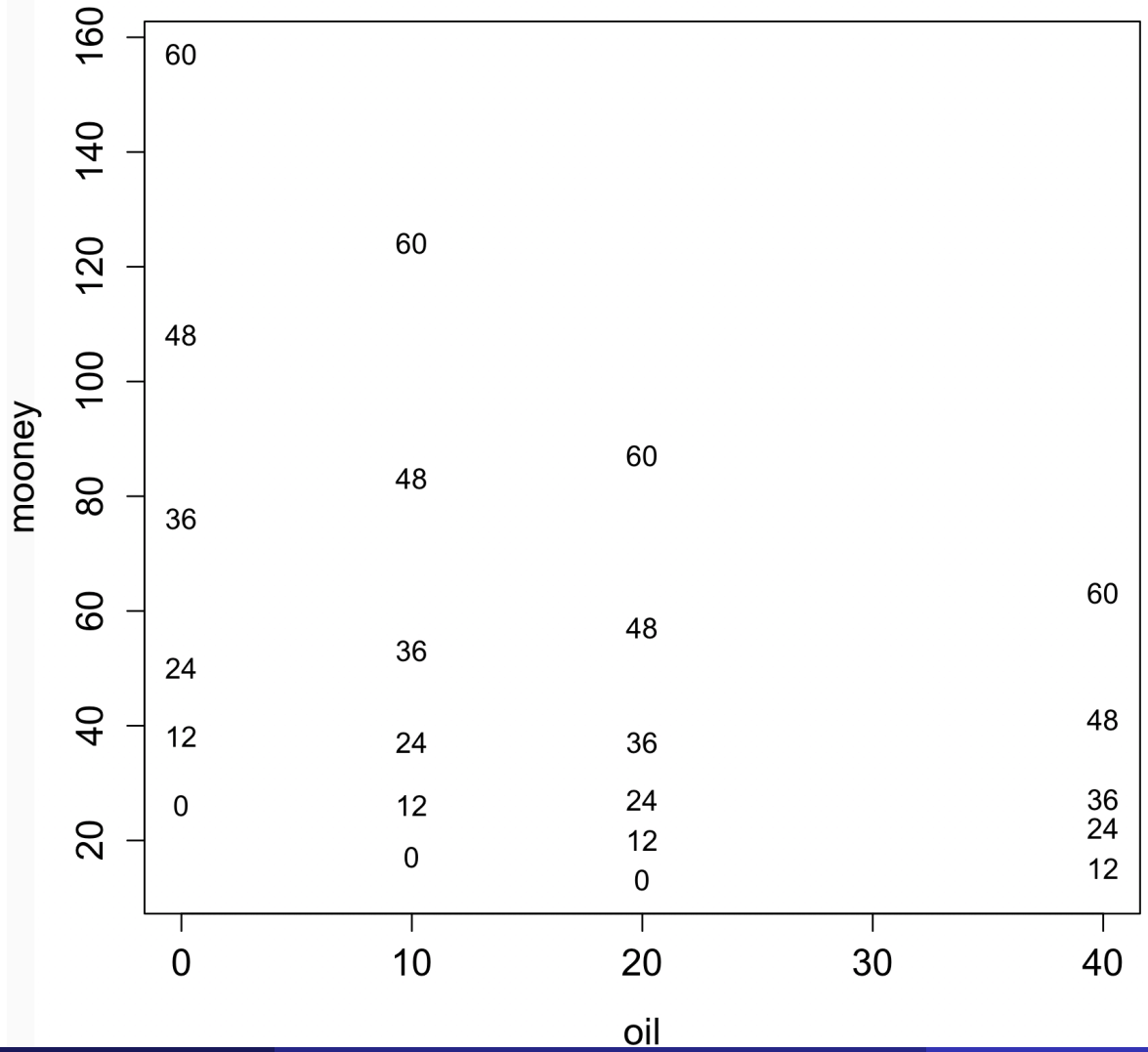
The plots can be generated using ggplot2() using

```
library(ggplot2)
p <- ggplot(mooney, aes(x = oil, y = mooney, label = filler))
p <- p + geom_text()
p <- p + scale_y_continuous(limits = c(0,
max(mooney$mooney, na.rm=TRUE)))
p <- p + labs(title="Mooney data, mooney by oil with
filler labels")
print(p)
## Warning: Removed 1 rows containing missing values (geom text).
library(ggplot2)
p <- ggplot(mooney, aes(x = filler, y = mooney, label = oil))
p <- p + geom_text()
```

Plots like these can also be made in base graphics by plotting an empty plot and then using the `text()` command, which is usually used to annotate graphs:

```
> attach(x)
> plot(oil,mooney,type="n",cex.lab=1.3,cex.axis=1.3)
> text(oil,mooney,as.character(filler))
```

Polynomial regression: cubic fit



From the plots, the relationship between viscosity and both variables appears to be curvilinear. This suggests adding quadratic terms for both variables.

```
> oil2 <- oil^2
> filler2 <- filler^2
> oil.m <- lm(mooney ~ oil+filler+oil2+filler2+oil*filler)
> summary(oil.m)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 27.144582   2.616779  10.373 9.02e-09 ***
oil         -1.271442   0.213533  -5.954 1.57e-05 ***
filler       0.436984   0.152658   2.862   0.0108 *
oil2     0.033611   0.004663   7.208 1.46e-06 ***
filler2  0.027323   0.002410  11.339 2.38e-09 ***
oil:filler  -0.038659   0.003187 -12.131 8.52e-10 ***
---
Signif. codes:  0 ?***? 0.001 ?**? 0.01 ?*? 0.05 ?.? 0.1 ? ? 1

Multiple R-squared:  0.9917,Adjusted R-squared:  0.9892
F-statistic: 405.2 on 5 and 17 DF,  p-value: < 2.2e-16
```

You can also use

```
lm.m.o2.f2 <- lm(mooney ~ oil + filler + I(oil^2) +
 I(filler^2) + I(oil * filler), data = mooney)
summary(lm.m.o2.f2)

# I() is used to create an interpreted object treated
# "as is", so we can include quadratic and cubic terms in
# the formula without creating separate columns in the
#dataset of these terms
```

From the output, the regression equation is

$$\begin{aligned}\text{Mooney} = 27.144 &- 1.271 \times \text{oil} + 0.437 \times \text{filler} \\ &+ 0.034 \times \text{oil}^2 + 0.027 \times \text{filler}^2 - 0.0387 \times \text{oil} \times \text{filler}\end{aligned}$$

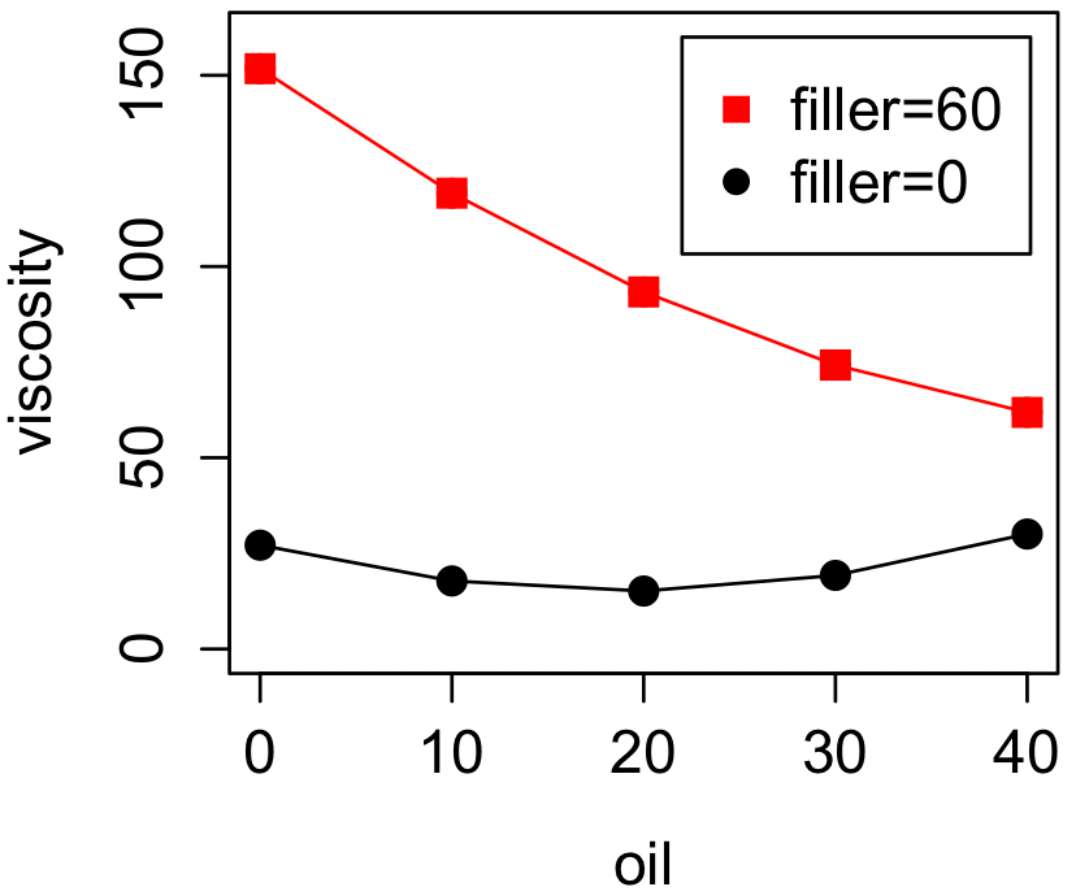
To see how the interaction works, let's predict some values.

```
> newoil <- c(0,10,20,30,40)
> newfiller <- c(0,0,0,0,0,60,60,60,60,60)
> newoil <- c(newoil,newoil)
> newoil2 <- newoil^2
> newfiller2 <- newfiller^2
> mydata <- data.frame(cbind(newoil,newfiller,
newoil2,newfiller2))
> names(mydata) <- c("oil","filler","oil2","filler2")
> a <- predict(oil.m,mydata)
> a
       1        2        3        4        5        6
27.14458 17.79121 15.15996 19.25082 30.06379 151.72512 119.1
       9       10
74.24519 61.86277
```

```
> plot(mydata$oil[1:5],a[1:5],pch=16,cex=1.5,cex.lab=1.3,
cex.axis=1.3,ylim=c(0,160),xlab="oil",ylab="viscosity")
> points(mydata$oil[1:5],a[1:5],type="l")
> points(mydata$oil[1:5],a[6:10],type="l",col="red")
> points(mydata$oil[1:5],a[6:10],pch=16,cex=1.5,
col="red")
> legend(22,160,legend=c("filler=60","filler=0"),
col=c("red","black"),pch=c(16,15),cex=1.3)
```

If you forget to match the variable names, here is the error you get (I often forget to match the names):

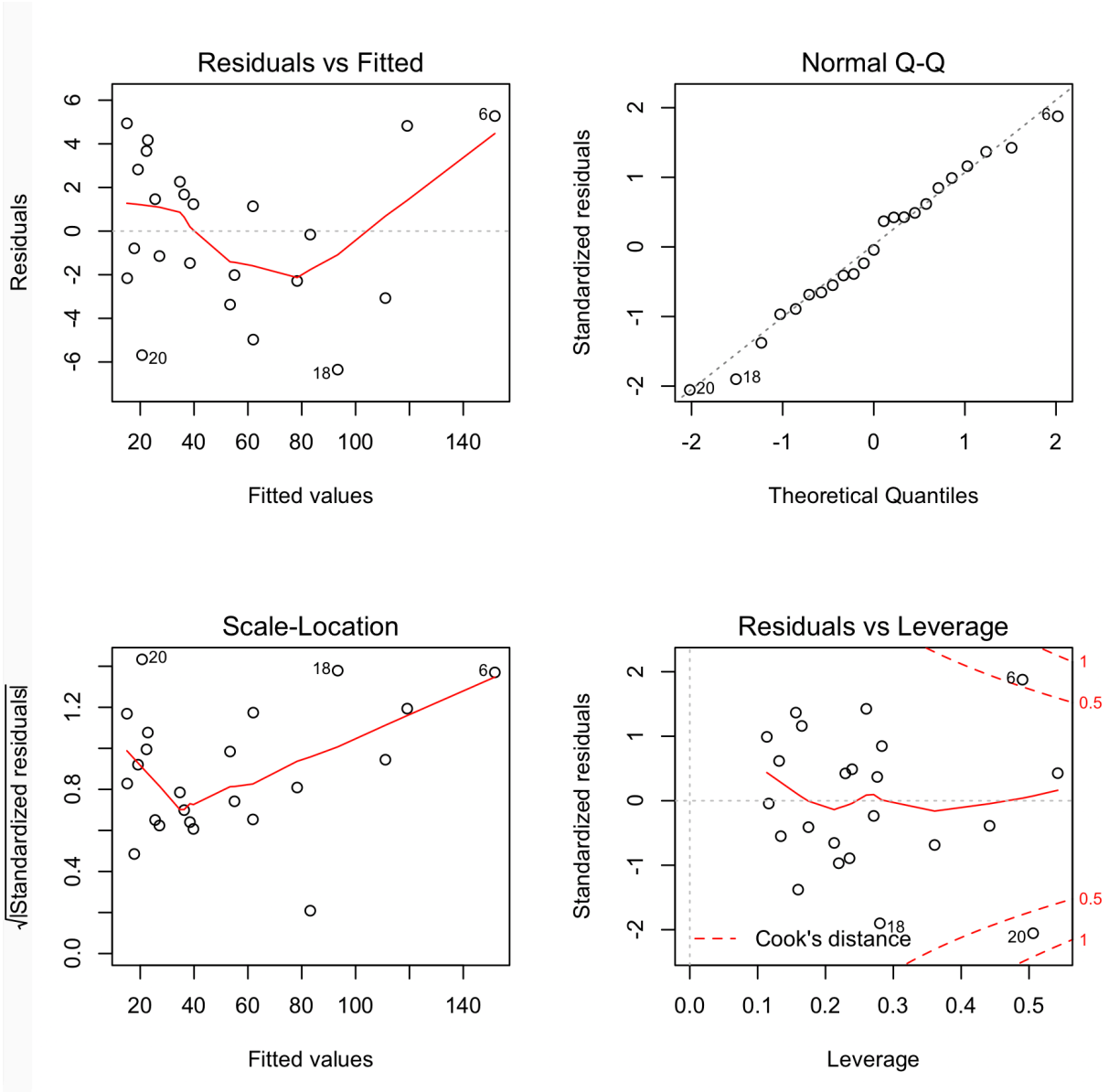
```
>  mydata <- data.frame(cbind(newoil,newfiller,newoil2,
newfiller2))
> a <- predict(oil.m,mydata)
Warning message:
'newdata' had 10 rows but variables found have 24 rows
```


viscosity
0
50
100
150
0
10
20
30
40
oil
filler=60
filler=0

Comments:

- all second-order terms, including interactions, are highly significant.
- R^2 values are extremely high, suggesting that not much else (for example cubic terms) would explain more of the response.
- The direction of the effects is hard to interpret because the signs change. —— For example, the effect of oil decreases viscosity in the first order term, but increases for the second order term, and the interaction is also negative, suggesting that as oil level increases, increasing the filler will decrease viscosity more, and vice versa (as filler increases, increasing oil decreases viscosity more.
- The plot helps illustrate the idea of the interaction. The relationship between viscosity and oil is quadratic for both levels of filler (I only plotted the two extreme values for the filler), but this quadratic relationship depends on the level of the filler. Similarly, one could make a plot of viscosity versus filler, and find that the quadratic relationship depends on the oil value.

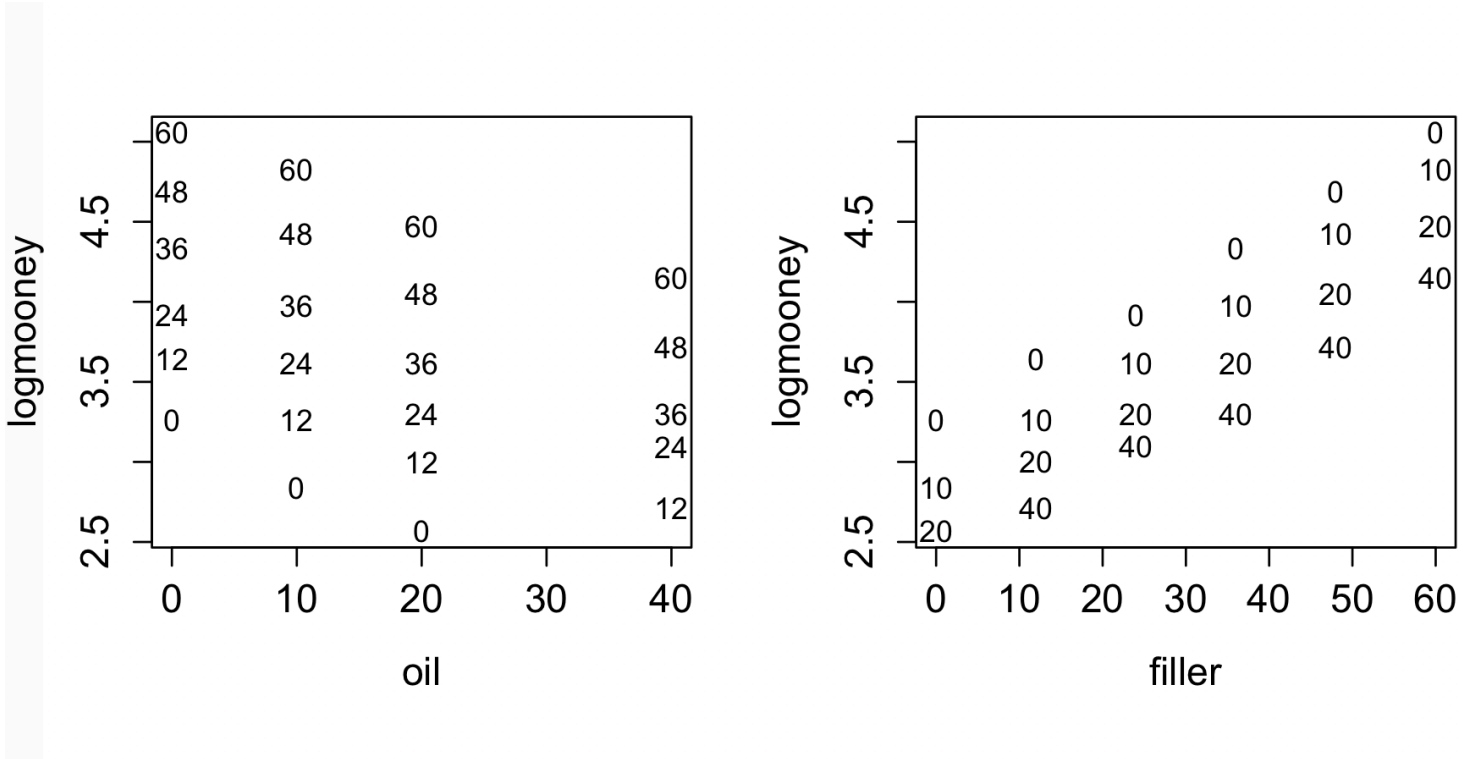
Residual plots



The residual plots look ok. There are potentially a couple of influential observations (points 6 and 20), but this does not seem bad.

Another possibility is to use the log of the Mooney viscosity. In this case, the log viscosity still seems to be quadratically related to oil, but linearly related to filler.

Log transformation on response



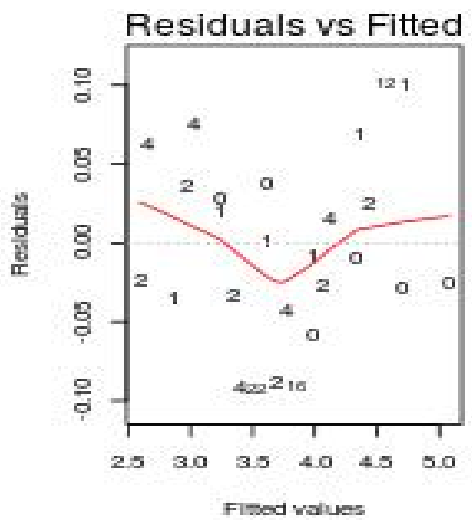
```
> oil.m2 <- lm(log(mooney) ~ oil + filler + oil2 +
filler2 + oil*filler)
> summary(oil.m2)
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.236e+00  3.557e-02  90.970  < 2e-16 ***
oil         -3.921e-02  2.903e-03 -13.507 1.61e-10 ***
filler       2.860e-02  2.075e-03  13.781 1.18e-10 ***
oil2         4.227e-04  6.339e-05   6.668 3.96e-06 ***
filler2      4.657e-05  3.276e-05   1.421    0.173
oil:filler  -4.231e-05  4.332e-05  -0.977    0.342

Multiple R-squared:  0.9954,Adjusted R-squared:  0.9941
```

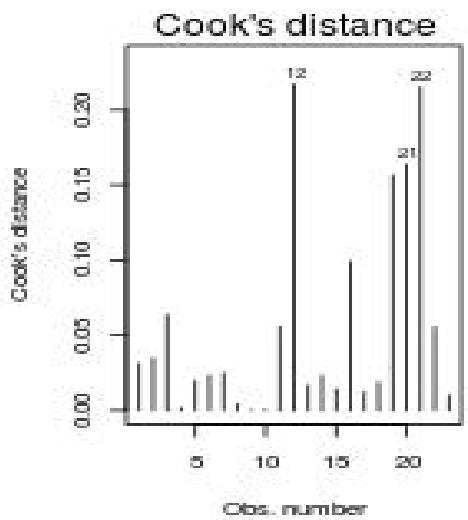
Here the interaction term isn't significant so we can remove it and refit the model. The quadratic term for filler is also not signifcant (after the interaction is removed), so we can remove that too.

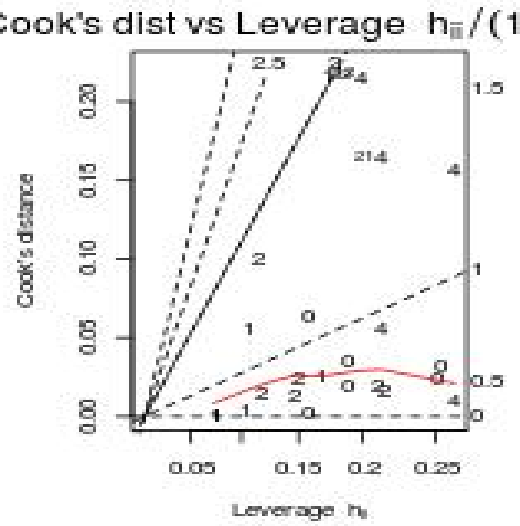
```
> oil.m3 <- lm(log(mooney) ~ oil + filler + oil2 + filler2 )
> oil.m3
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.251e+00  3.202e-02 101.537  < 2e-16 ***
oil         -4.033e-02  2.664e-03 -15.136 1.11e-11 ***
filler       2.838e-02  2.061e-03  13.773 5.32e-11 ***
oil2         4.146e-04  6.277e-05   6.605 3.34e-06 ***
filler2      3.997e-05  3.201e-05   1.248    0.228
```

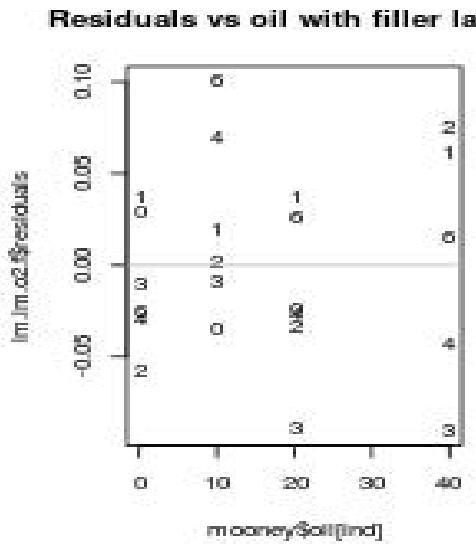
```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.230e+00  2.734e-02 118.139  < 2e-16 ***
oil         -4.024e-02  2.702e-03 -14.890 6.26e-12 ***
filler       3.086e-02  5.716e-04  53.986  < 2e-16 ***
oil2         4.097e-04  6.356e-05   6.446 3.53e-06 ***
---
Multiple R-squared:  0.9947,Adjusted R-squared:  0.9939
```

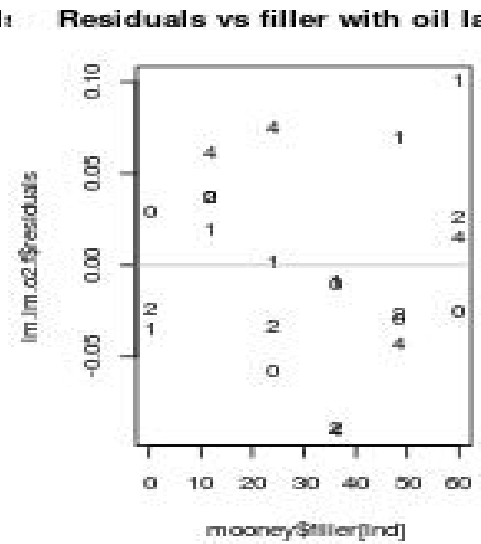
Diagnostics plots, looks good

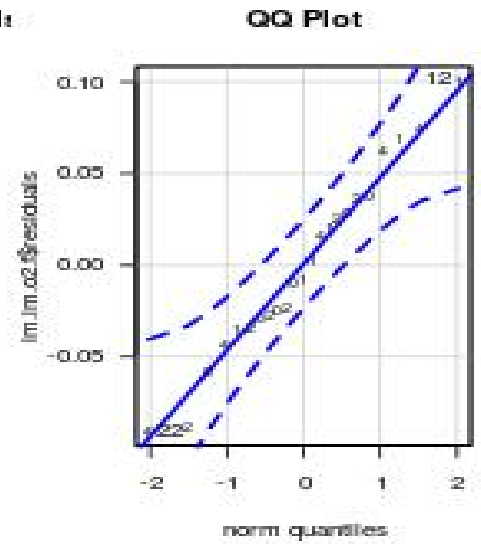












Both the full quadratic model and the model with log-transformed responses fit the data very well in terms of R^2 and adjusted R^2. There are pros and cons for the two models.

- Pros for the log-viscosity model are that there are fewer parameters and that it doesn't have an interaction term, making it easier to interpret.
- A pro for the quadratic model is that uses the original measurement scale, which again makes it easier to interpret in another sense, especially if you are using it to make predictions.

The predicted log Moody viscosity is given by

$$\widehat{log(Moody viscosity)} = 3.2297 - 0.0402 Oil + 0.0004 Oil^2 + 0.0309 Filler.$$

- Quadratic models with two or more predictors are often used in industrial experiments to estimate the optimal combination of predictor values to maximize or minimize the response, over the range of predictor variable values where the model is reasonable.
  —-(This strategy is called response surface methodology.)
  ——For example, we might wish to know what combination of oil level between 0 and 40 and filler level between 0 and 60 provides the lowest predicted Mooney viscosity (on the original or log scale). We can visually approximate the minimizer using the data plots, but one can do a more careful job of analysis using standard tools from calculus