Section 3.4 New Predictors from Old

Load needed packages.

```
library(Stat2Data)
library(mosaic)
library(ggplot2)
library(dplyr)
```

Create a dataframe for **Perch** and look at the structure of the data.

```
data("Perch")
str(Perch)
```

```
## 'data.frame':    56 obs. of  4 variables:
##  $ Obs   : int  104 105 106 107 108 109 110 111 112 113 ...
##  $ Weight: num  5.9 32 40 51.5 70 100 78 80 85 85 ...
##  $ Length: num  8.8 14.7 16 17.2 18.5 19.2 19.4 20.2 20.8 21 ...
##  $ Width : num  1.4 2 2.4 2.6 2.9 3.3 3.1 3.1 3 2.8 ...
```

EXAMPLE 3.11 Perch weights

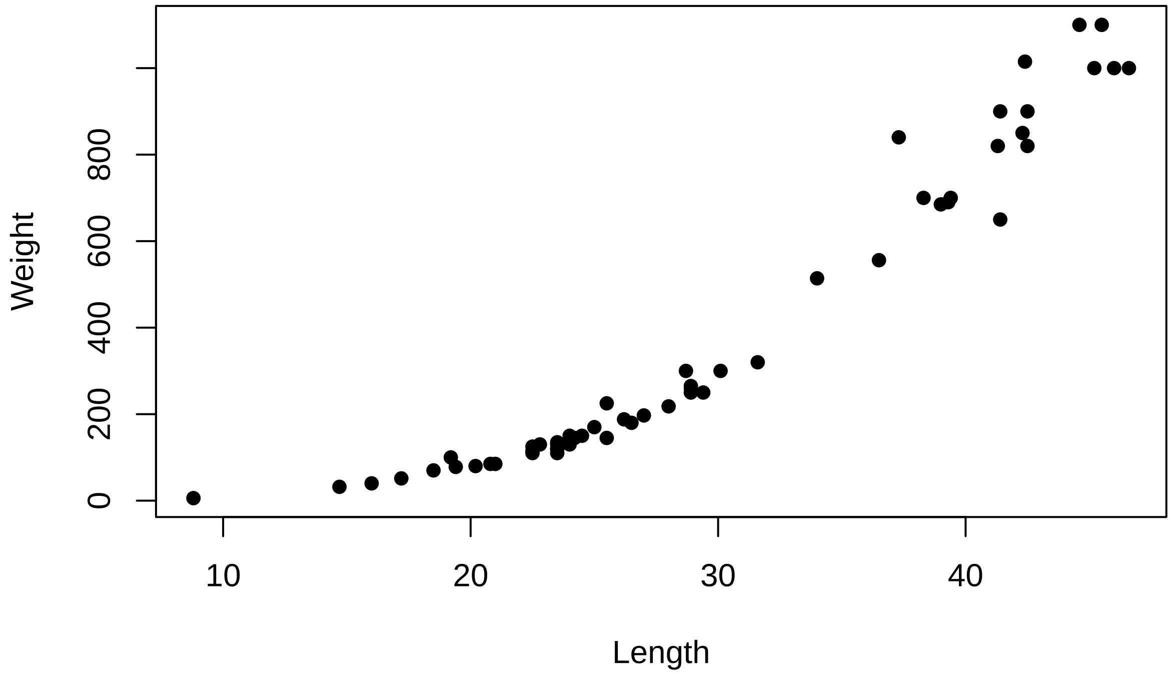
TABLE 3.3 First few cases of fish measurements in the **Perch** datafile

```
head(Perch)
```

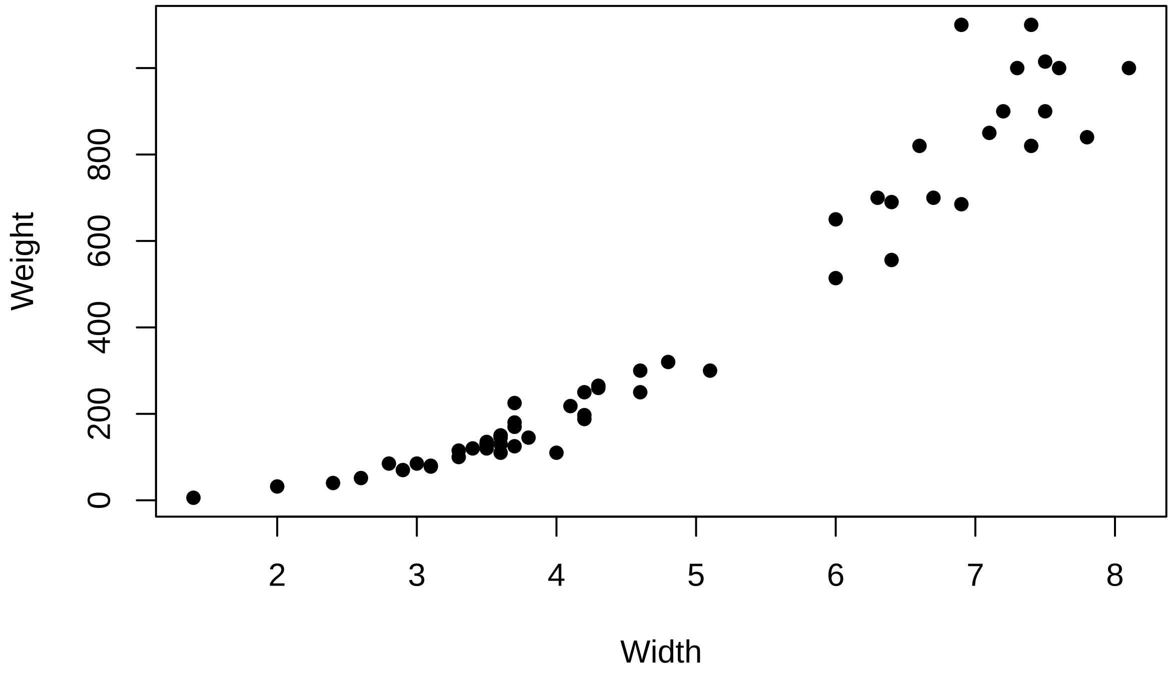
```
##   Obs Weight Length Width
## 1 104    5.9    8.8   1.4
## 2 105   32.0   14.7   2.0
## 3 106   40.0   16.0   2.4
## 4 107   51.5   17.2   2.6
## 5 108   70.0   18.5   2.9
## 6 109  100.0   19.2   3.3
```

FIGURE 3.12 Indiviudal predictors for perch weights

```
plot(Weight~Length,data=Perch,pch=16)
```



```
plot(Weight~Width,data=Perch,pch=16)
```



EXAMPLE 3.11 FIT the model with interaction alone

```
Perch$LengthxWidth=Perch$Length*Perch$Width
modint=lm(Weight~LengthxWidth,data=Perch)
summary(modint)
```

```
##
## Call:
## lm(formula = Weight ~ LengthxWidth, data = Perch)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -99.840 -26.148  -7.595  16.784 215.449
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -136.92622   12.72795  -10.76 4.83e-15 ***
## LengthxWidth    3.31929    0.06804   48.79  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.25 on 54 degrees of freedom
## Multiple R-squared:  0.9778, Adjusted R-squared:  0.9774
## F-statistic:  2380 on 1 and 54 DF,  p-value: < 2.2e-16
```

FIGURE 3.13 Perch weights plotted against the product of length and width

```
plot(Weight~LengthxWidth,data=Perch,pch=16)
```

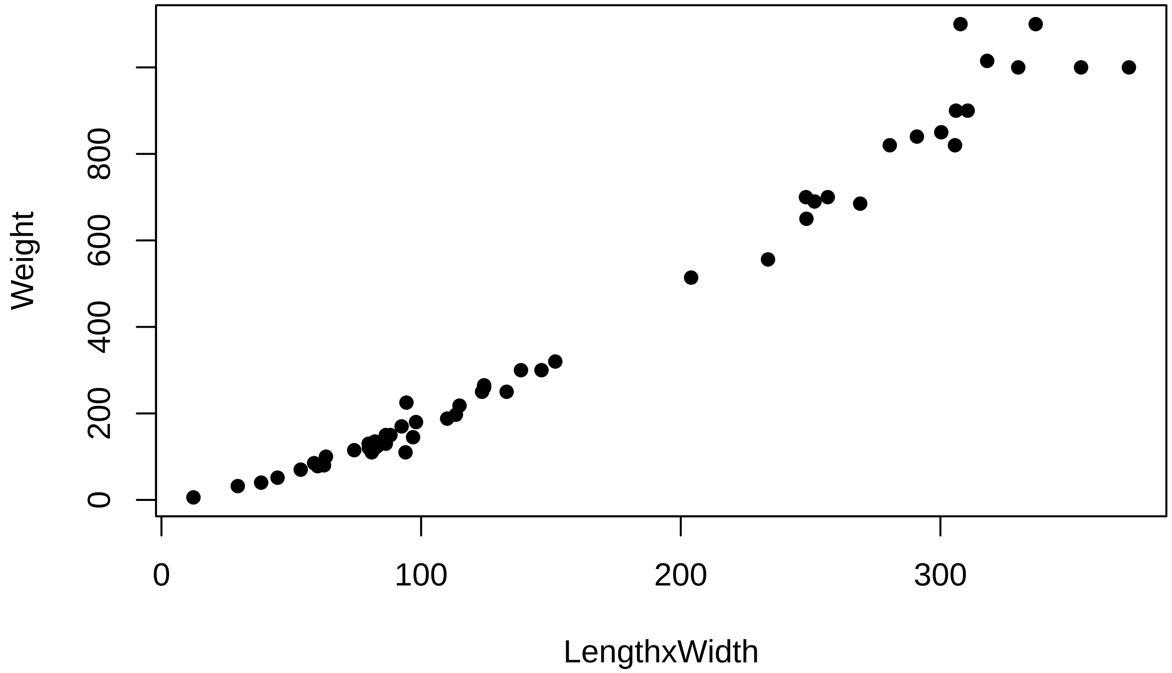
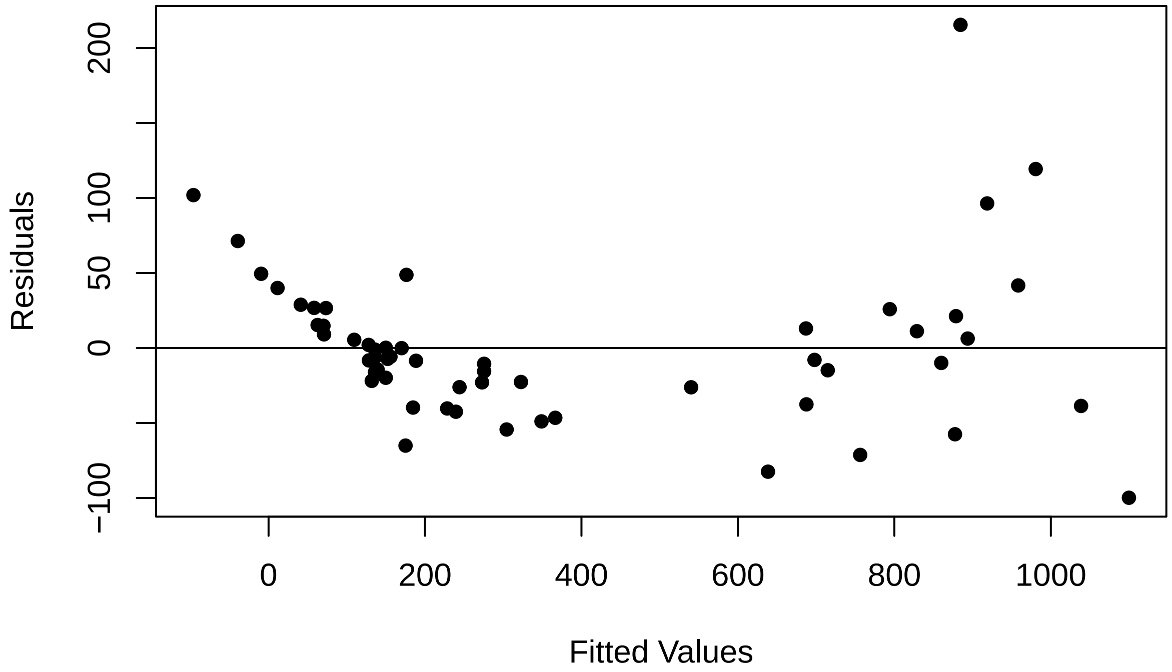


FIGURE 3.14 Residual plot for the one-term interaction model for perch weights

```
plot(modint$residuals~modint$fitted,xlab="Fitted Values",ylab="Residuals",pch=16)
abline(0,0)
```



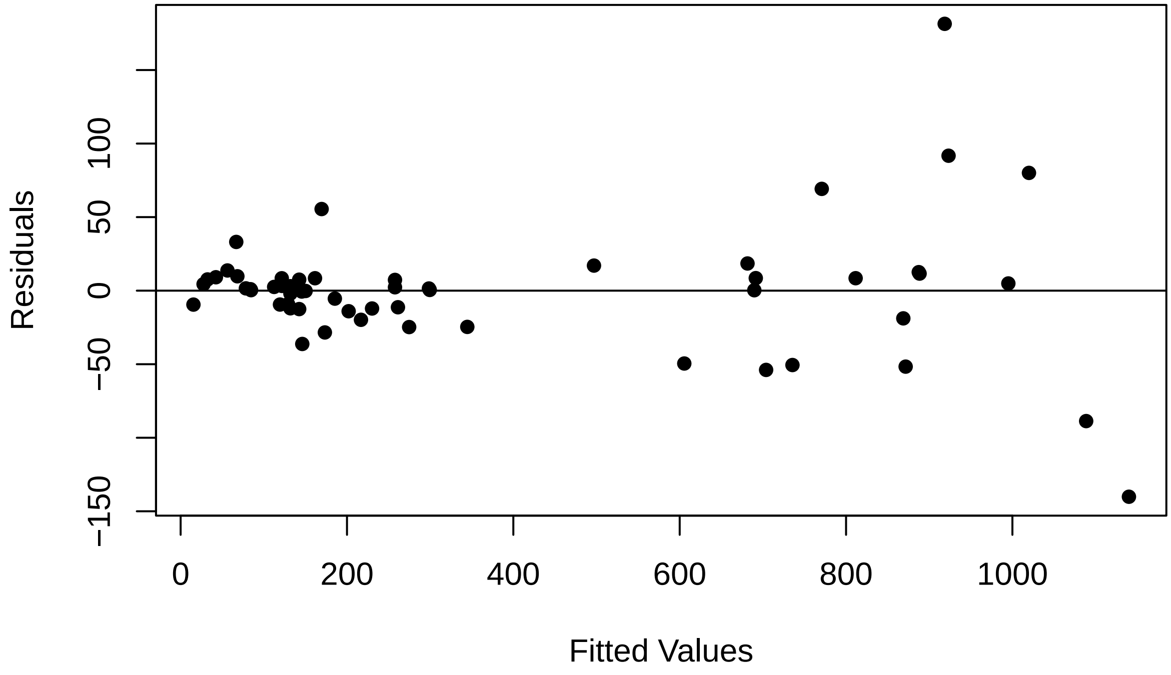
EXAMPLE 3.11 FIT the full interaction model

```
modintfull=lm(Weight~Length+Width+LengthxWidth,data=Perch)
summary(modintfull)
```

```
##
## Call:
## lm(formula = Weight ~ Length + Width + LengthxWidth, data = Perch)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -140.106  -12.226    1.230    8.489  181.408
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  113.9349    58.7844   1.938    0.058 .
## Length        -3.4827     3.1521  -1.105    0.274
## Width        -94.6309    22.2954  -4.244 9.06e-05 ***
## LengthxWidth   5.2412     0.4131  12.687  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.24 on 52 degrees of freedom
## Multiple R-squared:  0.9847, Adjusted R-squared:  0.9838
## F-statistic:  1115 on 3 and 52 DF,  p-value: < 2.2e-16
```

FIGURE 3.15 Residual plot for the full interaction model

```
plot(modintfull$residuals~modintfull$fitted,xlab="Fitted Values",ylab="Residuals",pch=16)
abline(0,0)
```



EXAMPLE 3.12 Guessing IQ

Create a dataframe for **IQGuessing** and look at the structure of the data.

```
data("IQGuessing")
str(IQGuessing)
```

```
## ’data.frame’:    40 obs. of  3 variables:
##  $ Age    : int  20 20 21 19 22 20 19 20 23 20 ...
##  $ GuessIQ: int  134 127 135 125 126 151 101 142 143 126 ...
##  $ TrueIQ : int  83 121 114 129 111 116 117 108 129 134 ...
```

TABLE 3.4 First six cases of age and IQ data in **IQGuessing**

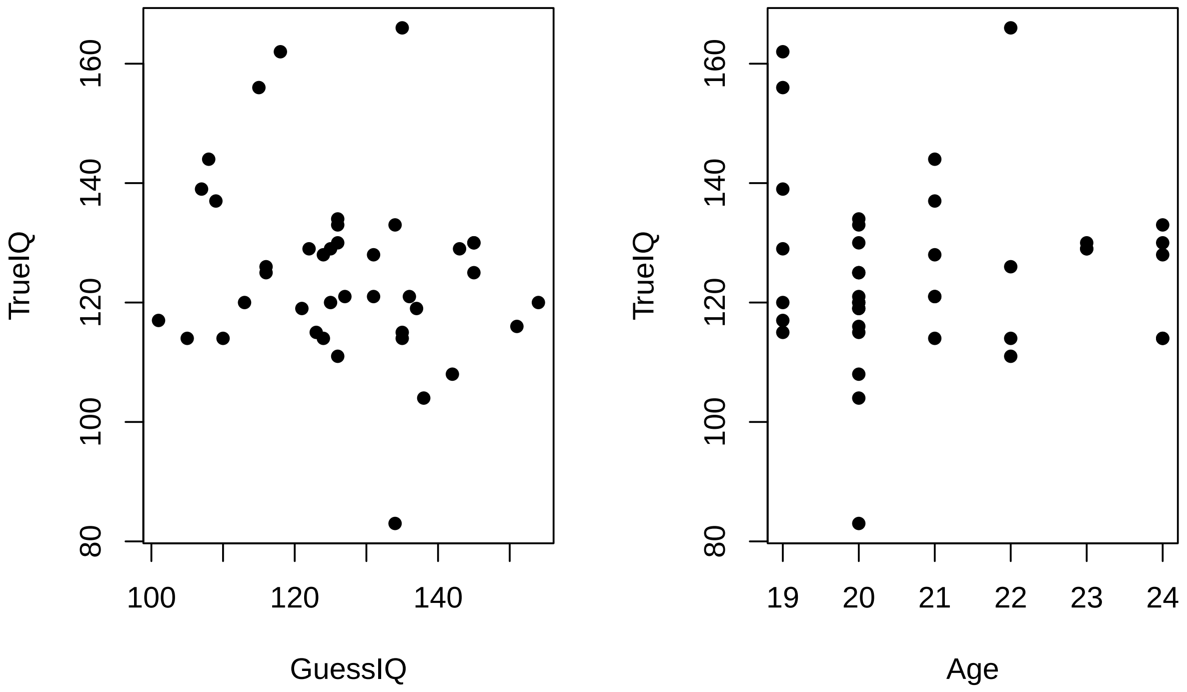
```
head(IQGuessing)
```

```
##   Age GuessIQ TrueIQ
## 1  20     134     83
## 2  20     127    121
## 3  21     135    114
## 4  19     125    129
## 5  22     126    111
## 6  20     151    116
```

FIGURE 3.16 Scatterplots of individual predictors of TrueIQ

Note: The mfrow=c(1,2) tells R to create a plot with two subplots, one for each scatterplot. We are providing this code just in case you want to create side-by-side plots.

```
par(mfrow=c(1,2))
par(mar=c(8,5,2,1))
plot(TrueIQ ~ GuessIQ, pch=16, data=IQGuessing)
plot(TrueIQ ~ Age, pch=16, data=IQGuessing)
```



```
layout(mat=c(1,1))
```

EXAMPLE 3.12 FIT a regression model predicting TrueIQ using GuessIQ for two age groups (20 and under, over 20)

Use ifelse() in dplyr to create age groups.

```
IQGuessing$Agegroup <- ifelse(IQGuessing$Age > 20.5, 1, 0)
```

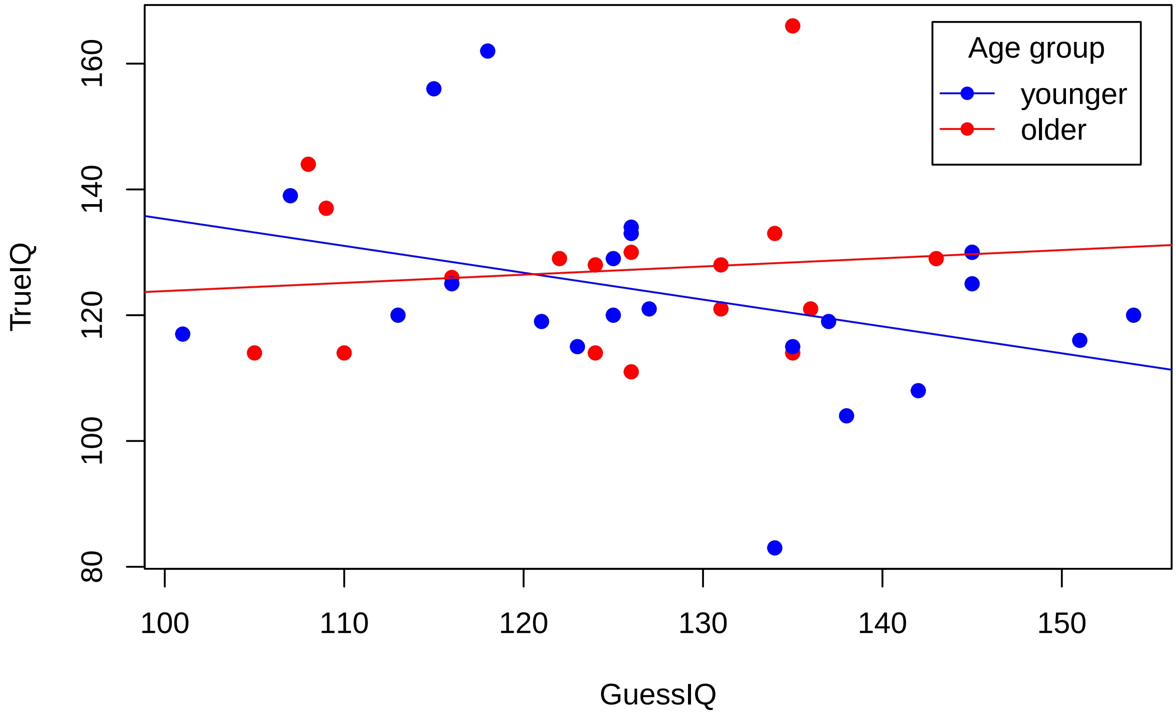
Note that TrueIQ~GuessIQ*Agegroup is an R trick for specifying putting both variables and their product interaction into the model.

```
model2=lm(TrueIQ ~ GuessIQ*Agegroup,data=IQGuessing)
model2
```

```
##
## Call:
## lm(formula = TrueIQ ~ GuessIQ * Agegroup, data = IQGuessing)
##
## Coefficients:
##      (Intercept)           GuessIQ          Agegroup  GuessIQ:Agegroup
##         177.9912           -0.4270          -67.1976            0.5574
```

FIGURE 3.17 Regression lines for predicting TrueIQ from GuessedIQ for younger and older women

```
plot(IQGuessing$TrueIQ~IQGuessing$GuessIQ,pch=19,col=c("blue","red")[IQGuessing$Agegroup+1], xlab="Gues
abline(model2$coef[1],model2$coef[2],col="blue")
abline(model2$coef[1]+model2$coef[3],(model2$coef[2]+model2$coef[4]),col="red")
legend("topright", inset=.03, title="Age group",c("younger","older"),lty=1,
       pch=16,col=c("blue","red"))
```



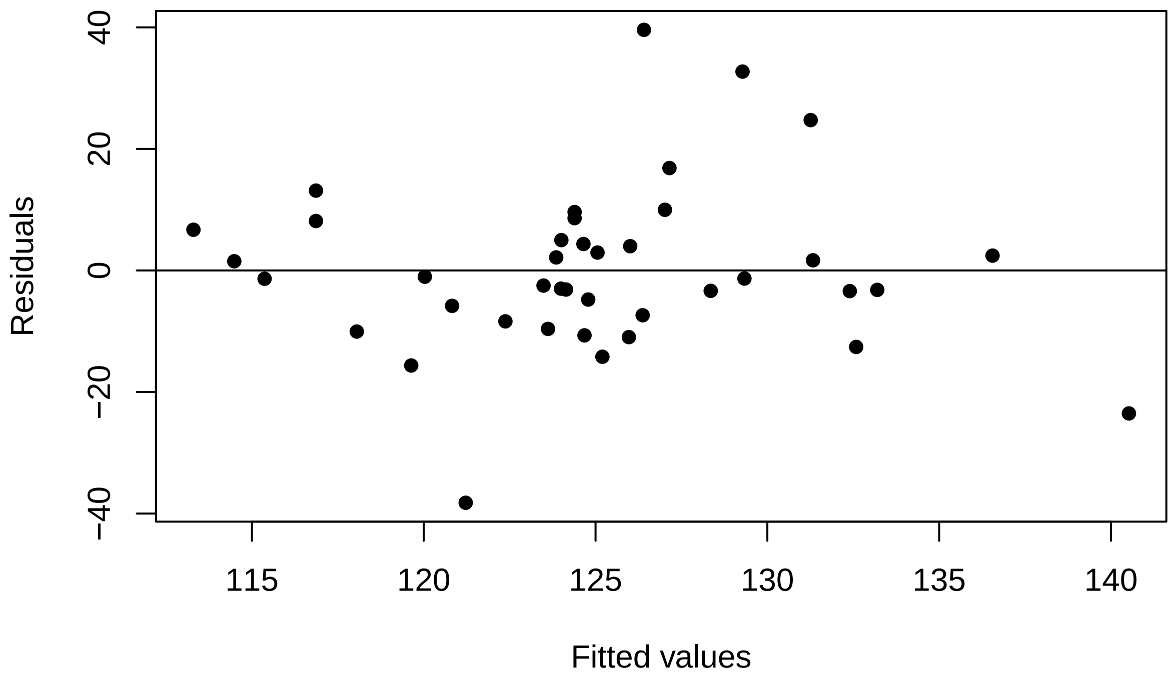
EXAMPLE 3.12 FIT a multiple regression model using Age (not the two groups) and GuessIR with interaction

```r
IQmodel <- lm(TrueIQ~GuessIQ+Age+GuessIQ*Age,data=IQGuessing)
summary(IQmodel)
```

```
##
## Call:
## lm(formula = TrueIQ ~ GuessIQ + Age + GuessIQ * Age, data = IQGuessing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.221  -7.622  -1.351   5.422  39.591
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 834.7621   320.9342   2.601   0.0134 *
## GuessIQ      -5.7021     2.5584  -2.229   0.0322 *
## Age         -33.0227    15.5534  -2.123   0.0407 *
## GuessIQ:Age   0.2653     0.1239   2.142   0.0390 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.39 on 36 degrees of freedom
## Multiple R-squared:  0.1502, Adjusted R-squared:  0.07939
## F-statistic: 2.121 on 3 and 36 DF,  p-value: 0.1146
```

FIGURE 3.18 Residual plot for the regression of TrueIQ on GuessIQ, Age, and their interaction

```r
plot(IQmodel$residuals~IQmodel$fitted.values,pch=16,xlab="Fitted values",ylab="Residuals")
abline(h=0)
```



EXAMPLE 3.12 FIT with no interaction term

```
IQmodelnoint <- lm(TrueIQ~GuessIQ+Age,data=IQGuessing)
summary(IQmodelnoint)
```

```
##
## Call:
## lm(formula = TrueIQ ~ GuessIQ + Age, data = IQGuessing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.215  -8.283  -1.826   7.791  42.714
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 151.6593    37.6397   4.029 0.000268 ***
## GuessIQ      -0.2351     0.1848  -1.272 0.211301
## Age           0.1532     1.4849   0.103 0.918409
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.08 on 37 degrees of freedom
## Multiple R-squared:  0.0419, Adjusted R-squared:  -0.009886
## F-statistic: 0.8091 on 2 and 37 DF,  p-value: 0.453
```

EXAMPLE 3.13 Daily carbon dioxide

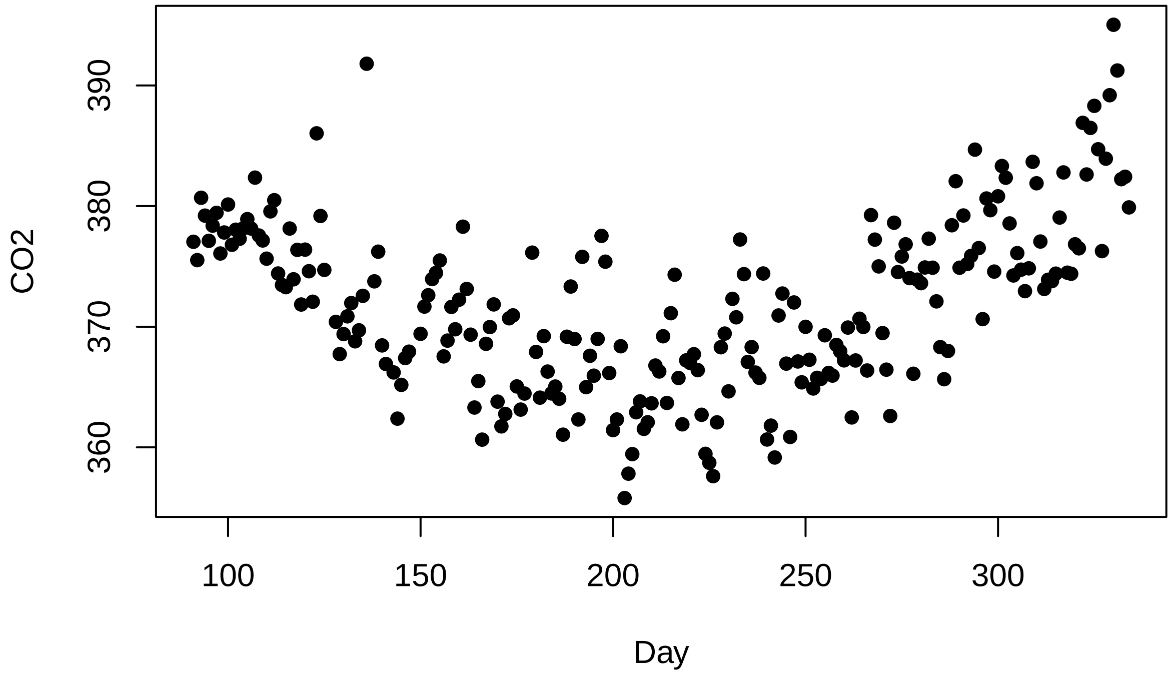
Create a dataframe for **CO2Germany** and look at the structure of the data.

```
data("CO2Germany")
str(CO2Germany)
```

```
## 'data.frame':    237 obs. of  2 variables:
##  $ CO2: num  377 376 381 379 377 ...
##  $ Day: int  91 92 93 94 95 96 97 98 99 100 ...
```

FIGURE 3.19 CO2 levels by day, April-November 2001

```
plot(CO2~Day,pch=16,data=CO2Germany)
```



EXAMPLE 3.13 FIT

Now for the quadratic model.

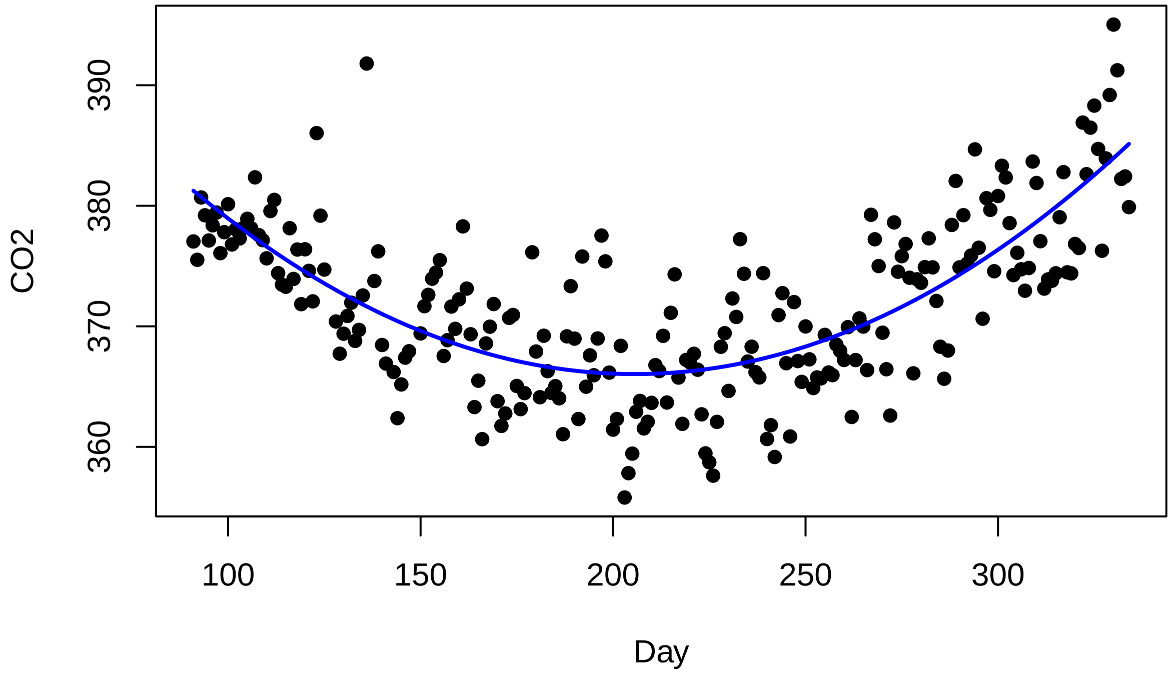
```
CO2Germany$DaySq <- CO2Germany$Day^2
CO2model <- lm(CO2~Day+DaySq, data=CO2Germany)
CO2model
```

```
##
## Call:
## lm(formula = CO2 ~ Day + DaySq, data = CO2Germany)
##
```

```
## Coefficients:
## (Intercept)          Day        DaySq
##  414.974747    -0.476034     0.001158
```

FIGURE 3.20 Quadratic regression fit for CO2 levels

```
plot(CO2~Day, pch=16,data=CO2Germany)
lines(CO2model$fitted~CO2Germany$Day,lwd=2,col="blue")
```



EXAMPLE 3.13 ASSESS

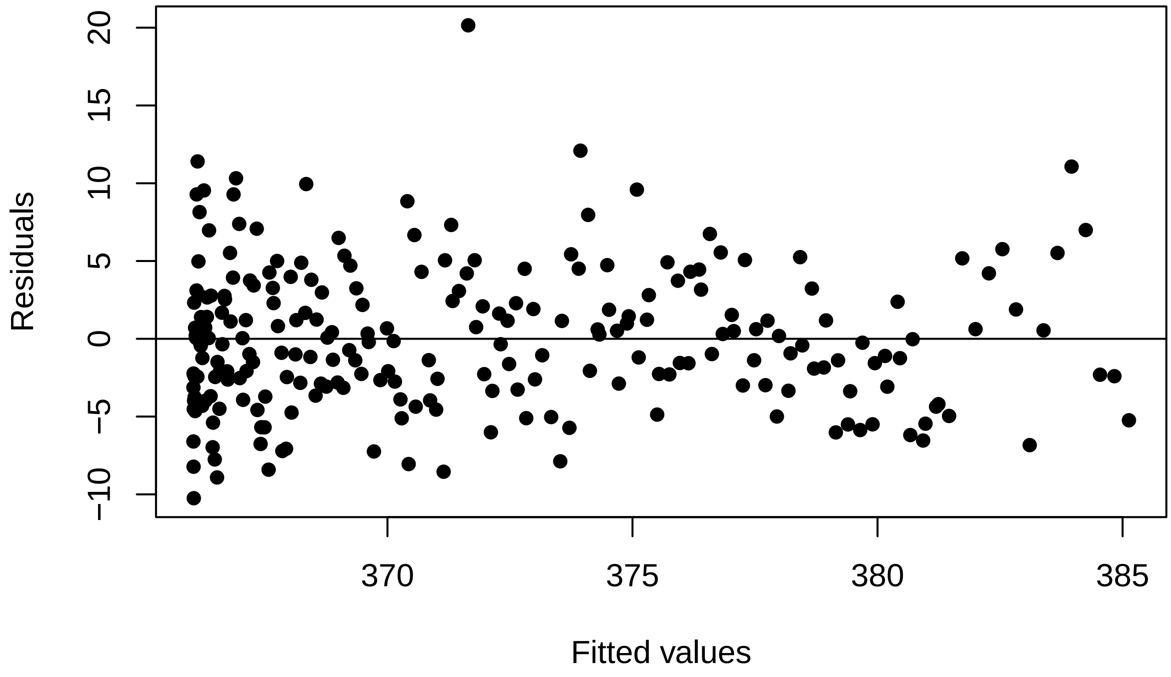
```
summary(CO2model)
```

```
##
## Call:
## lm(formula = CO2 ~ Day + DaySq, data = CO2Germany)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2482  -3.0799  -0.2524   2.8430  20.1527
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.150e+02  2.856e+00  145.28   <2e-16 ***
## Day         -4.760e-01  2.874e-02  -16.57   <2e-16 ***
## DaySq        1.158e-03  6.684e-05   17.32   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.619 on 234 degrees of freedom
## Multiple R-squared:  0.5734, Adjusted R-squared:  0.5698
## F-statistic: 157.3 on 2 and 234 DF,  p-value: < 2.2e-16
```

FIGURE 3.21 Diagnostic plots from quadratic regression fit for CO2 levels

(a) Residuals versus fits

```
plot(CO2model$residuals~CO2model$fitted,pch=16, ylab="Residuals",xlab="Fitted values")
abline(h=0)
```



(b) Normal quantile plot

```
qqnorm(CO2model$residuals, xlab="Normal Quantiles", ylab="Residuals",main="", pch=16)
qqline(CO2model$residuals)
```

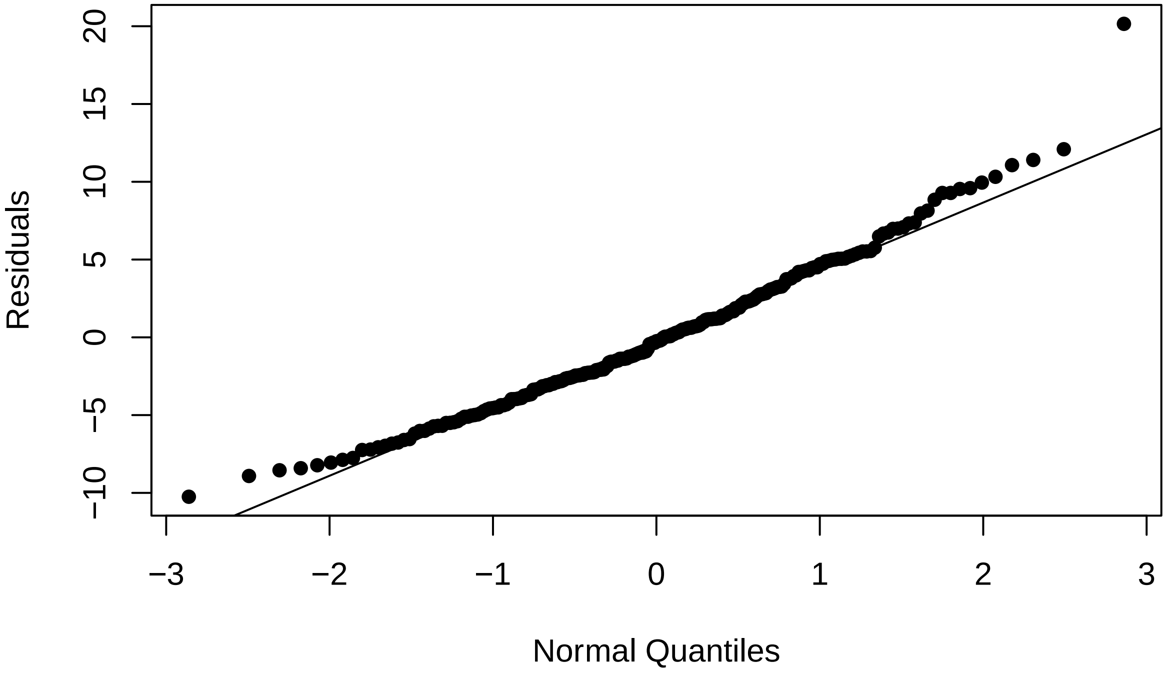
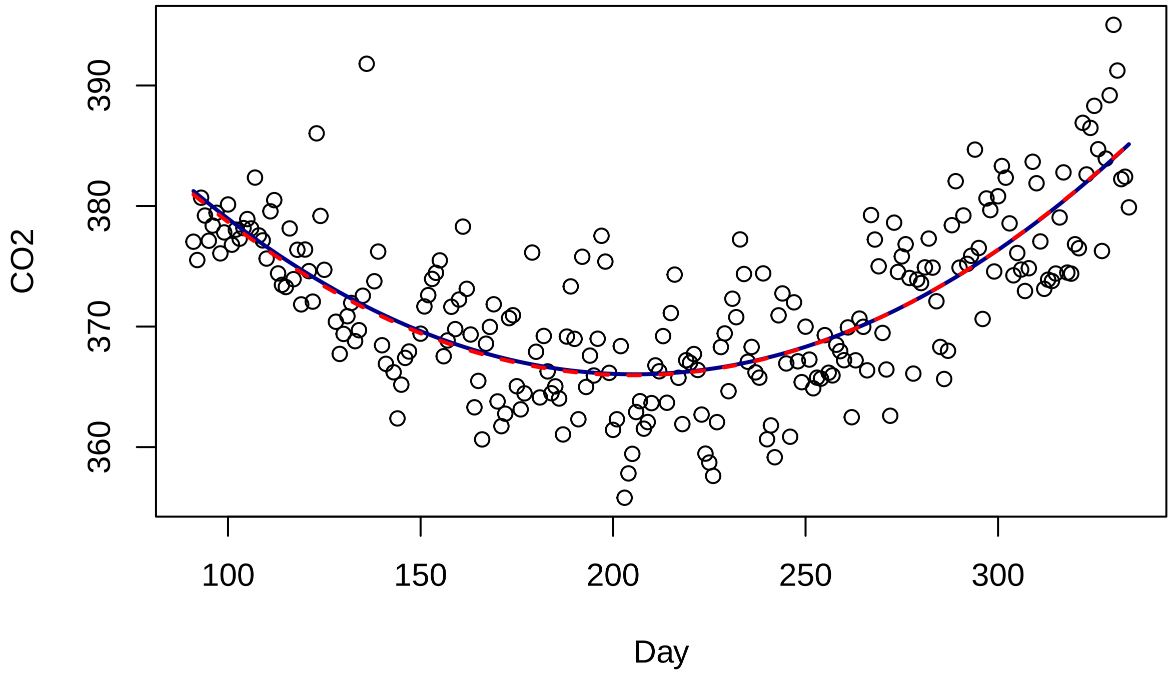


FIGURE 3.22 Quadratic regression fit for CO2 levels with all data (solid line) and with outlier removed (dashed line)

```r
NewCO2 <- subset(CO2Germany, CO2model$residuals < 15)
CO2modelNew <- lm(CO2~Day+DaySq, data=NewCO2)
plot(CO2~Day, data=CO2Germany)
lines(CO2model$fitted~CO2Germany$Day,col="darkblue",lwd=2)
lines(CO2modelNew$fitted~NewCO2$Day,lty=2,col="red",lwd=2)
```



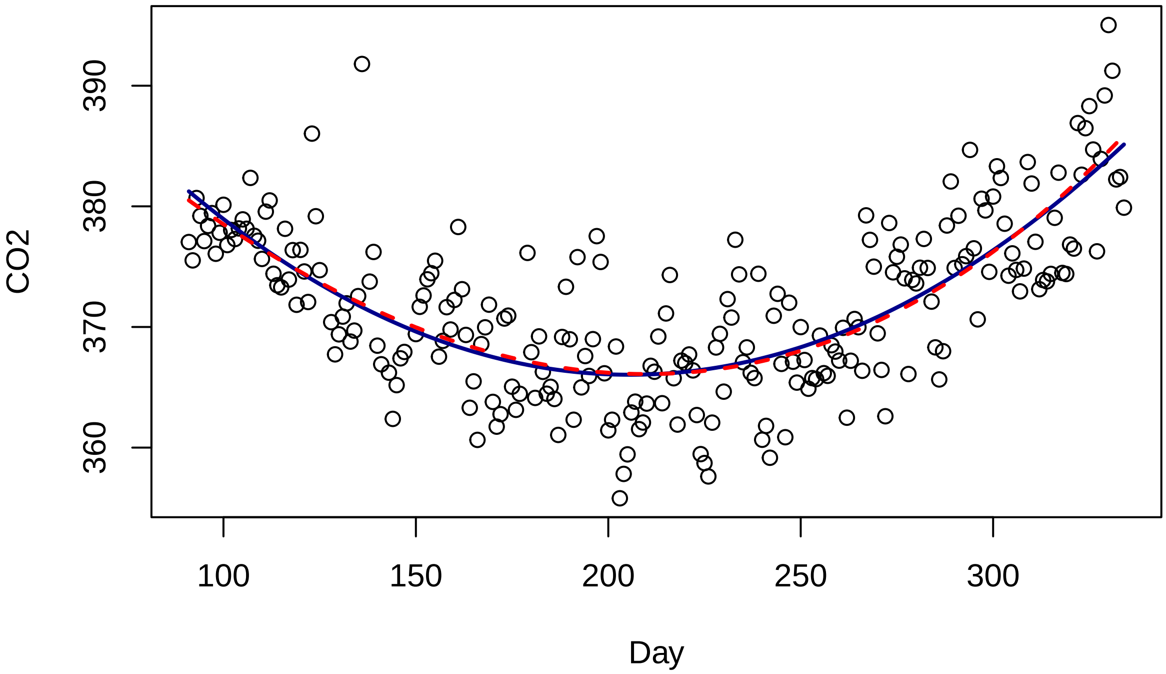
EXAMPLE 3.13 FIT cubic model

```
CO2Germany$Day3 <- CO2Germany$Day^3
CO2modelcubic <- lm(CO2~Day+DaySq+Day3, data=CO2Germany)
summary(CO2modelcubic)
```

```
##
## Call:
## lm(formula = CO2 ~ Day + DaySq + Day3, data = CO2Germany)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3483  -2.9931  -0.3974   2.8296  19.8833
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.067e+02  8.848e+00  45.963   <2e-16 ***
## Day         -3.396e-01  1.410e-01  -2.409   0.0168 *
## DaySq        4.703e-04  6.989e-04   0.673   0.5017
## Day3         1.078e-06  1.091e-06   0.988   0.3241
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.619 on 233 degrees of freedom
## Multiple R-squared:  0.5752, Adjusted R-squared:  0.5697
## F-statistic: 105.2 on 3 and 233 DF,  p-value: < 2.2e-16
```

FIGURE 3.23 Quadratic regression for CO2 levels (solid line) and cubic regression (dashed line)

```
plot(CO2~Day, data=CO2Germany)
lines(CO2model$fitted~CO2Germany$Day,col="darkblue",lwd=2)
lines(CO2modelcubic$fitted~CO2Germany$Day,col="red",lty=2,lwd=2)
```



EXAMPLE 3.14 Funnel swirling

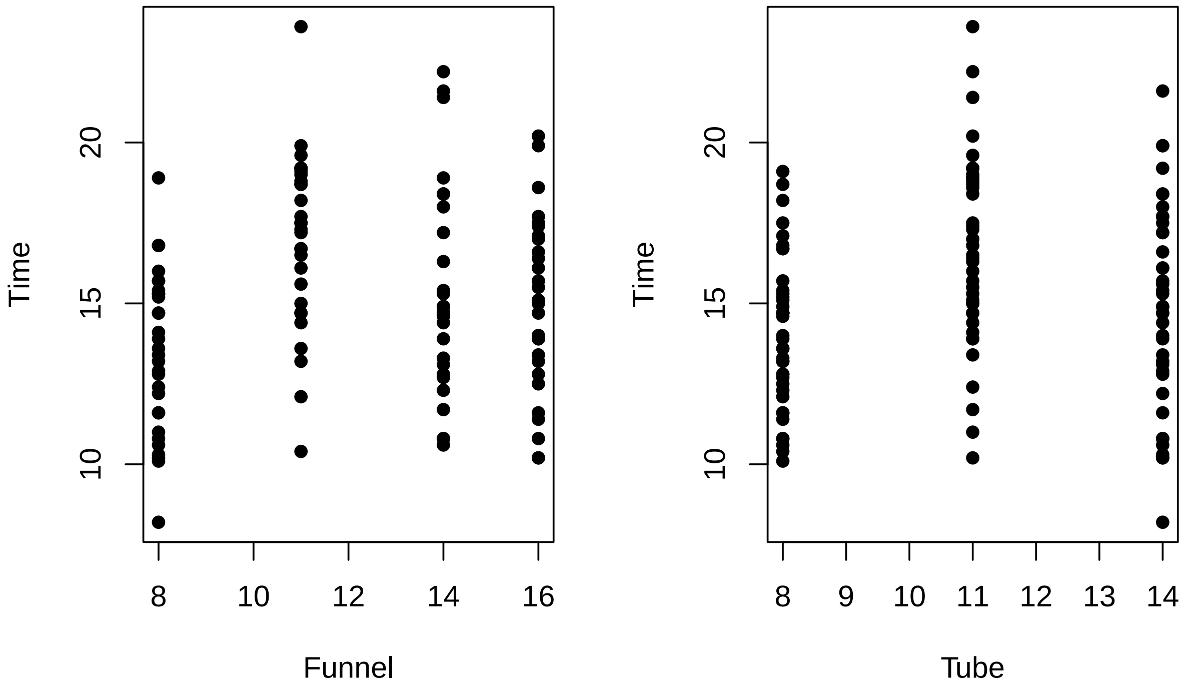
Create a dataframe for **FunnelDrop** and look at the structure of the data.

```
data("FunnelDrop")
str(FunnelDrop)
```

```
## 'data.frame':    120 obs. of  3 variables:
##  $ Funnel: int  8 8 8 8 8 8 8 8 8 8 ...
##  $ Tube  : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ Time  : num  15.3 15.2 14.7 11.6 15.7 11.6 10.1 16.8 15.4 13.6 ...
```

FIGURE 3.25 Scatterplots of drop times versus funnel and tube heights

```
par(mfrow=c(1,2))
par(mar=c(8,5,2,1))
plot(Time ~ Funnel, pch=16, data=FunnelDrop)
plot(Time ~ Tube, pch=16, data=FunnelDrop)
```



```
layout(mat=c(1,1))
```

EXAMPLE 3.14 FIT the complete second-order model

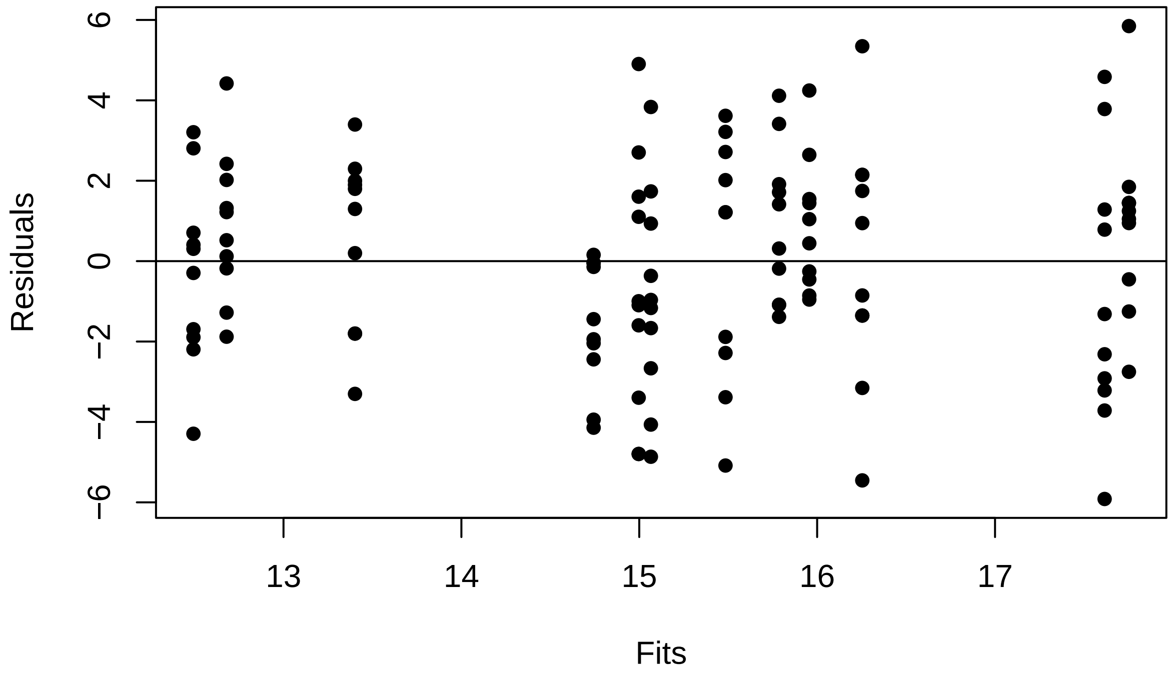
```
FunnelDrop$Funnelsq=FunnelDrop$Funnel^2
FunnelDrop$Tubesq=FunnelDrop$Tube^2
FunnelDrop$FunnelTube=FunnelDrop$Funnel*FunnelDrop$Tube
funnelmodel=lm(Time~Funnel+Tube+Funnelsq+Tubesq+FunnelTube, data=FunnelDrop)
summary(funnelmodel)
```

```
##
## Call:
## lm(formula = Time ~ Funnel + Tube + Funnelsq + Tubesq + FunnelTube,
##     data = FunnelDrop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9161 -1.7207  0.1383  1.7376  5.8472
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -26.80000    9.09938  -2.945 0.003913 **
## Funnel        3.13797    0.90140   3.481 0.000708 ***
## Tube          4.48722    1.28788   3.484 0.000701 ***
## Funnelsq     -0.15691    0.03455  -4.541 1.40e-05 ***
## Tubesq       -0.23528    0.05563  -4.229 4.75e-05 ***
## FunnelTube    0.06719    0.03179   2.114 0.036724 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.585 on 114 degrees of freedom
## Multiple R-squared:  0.2934, Adjusted R-squared:  0.2624
## F-statistic: 9.465 on 5 and 114 DF,  p-value: 1.433e-07
```

FIGURE 3.26 Residual plots for the second-order model for funnel drops

(a) Residuals versus fits

```
plot(funnelmodel$residuals~funnelmodel$fitted,pch=16, ylab="Residuals",xlab="Fits")
abline(h=0)
```



(b) Normal quantile plot

```
qqnorm(funnelmodel$residuals, xlab="Normal Quantiles", ylab="Residuals",main="", pch=16)
qqline(funnelmodel$residuals)
```

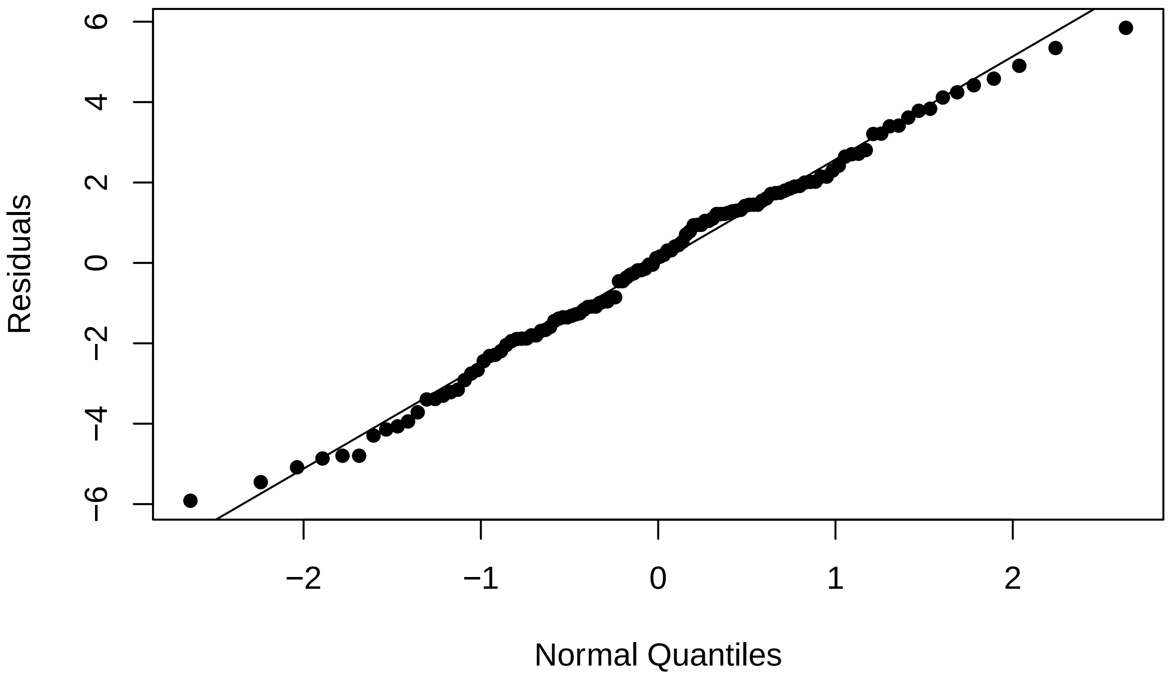
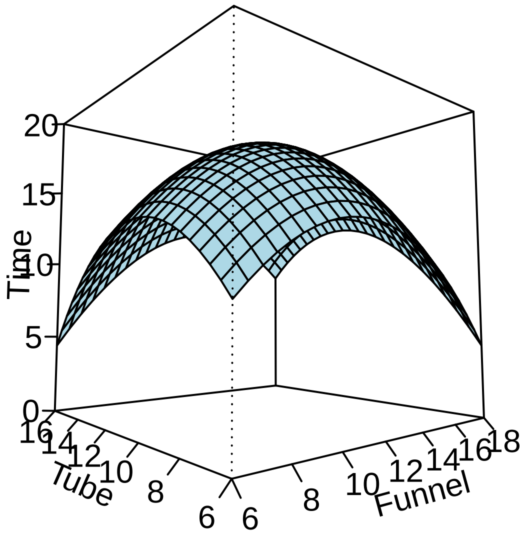


FIGURE 3.27 Predicted times from the fitted model for the funnel drop experiment

Plot to see the 3D surface

```
x=seq(6,18,length=25)
y=seq(6,16,length=25)
xsq=x^2
ysq=y^2
xy=x*y
z=outer(x,y,function(x,y){predict(funnelmodel,data.frame(Funnel=x,Tube=y,Funnelsq=x^2,Tubesq=y^2,Funnel
p=persp(x,y,z,theta=-40,phi=-5,col="lightblue",ticktype="detailed",xlab="Funnel",ylab="Tube",zlim=c(0,2
```



Alternate Solution

EXAMPLE 3.14 FIT the complete second-order model

We can add a function of an existing variable to a regression model by putting an I( ) around it and avoid needing to create new variables for terms like squares and products. The second-order model for the funnel experiment can be specified as below.

```
mod2a=lm(Time~Funnel+Tube+I(Funnel^2)+I(Tube^2)+I(Funnel*Tube),data=FunnelDrop)
summary(mod2a)
```

```
##
## Call:
## lm(formula = Time ~ Funnel + Tube + I(Funnel^2) + I(Tube^2) +
##     I(Funnel * Tube), data = FunnelDrop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9161 -1.7207  0.1383  1.7376  5.8472
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -26.80000    9.09938  -2.945 0.003913 **
## Funnel             3.13797    0.90140   3.481 0.000708 ***
## Tube               4.48722    1.28788   3.484 0.000701 ***
## I(Funnel^2)       -0.15691    0.03455  -4.541 1.40e-05 ***
## I(Tube^2)         -0.23528    0.05563  -4.229 4.75e-05 ***
## I(Funnel * Tube)   0.06719    0.03179   2.114 0.036724 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.585 on 114 degrees of freedom
## Multiple R-squared:  0.2934, Adjusted R-squared:  0.2624
## F-statistic: 9.465 on 5 and 114 DF,  p-value: 1.433e-07
```

Note: We could drop the I( ) from the product term and it would still work.