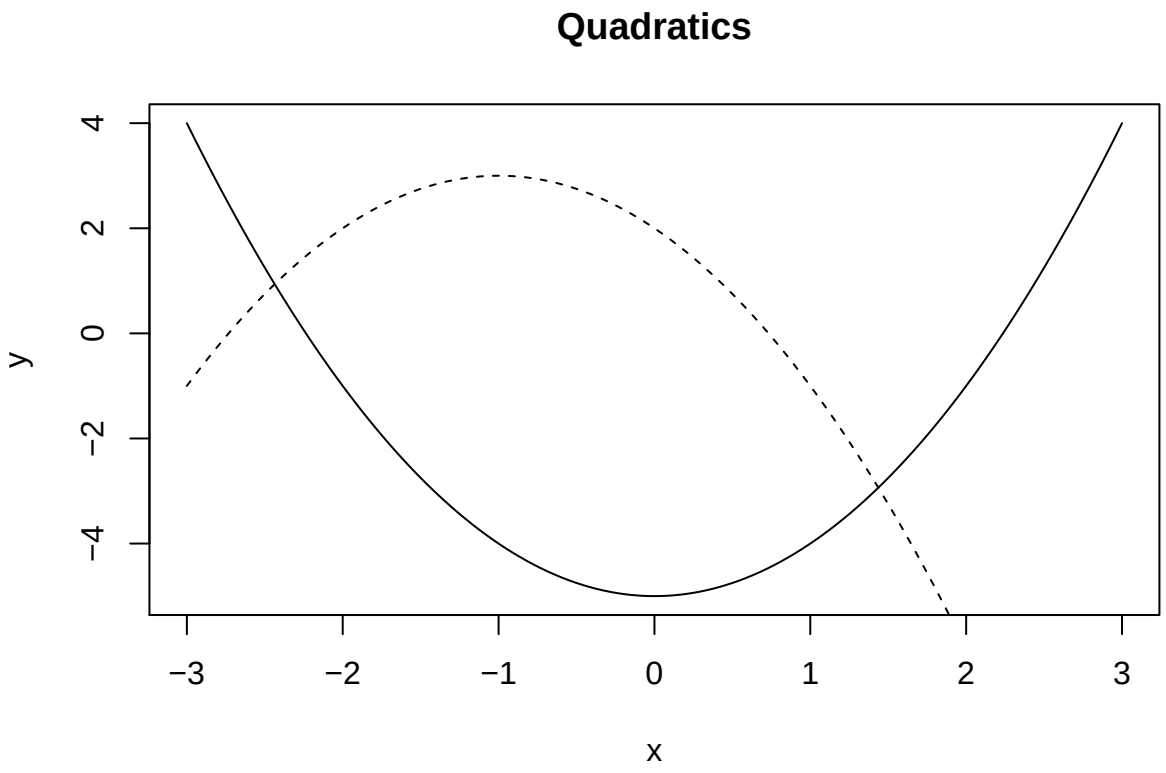
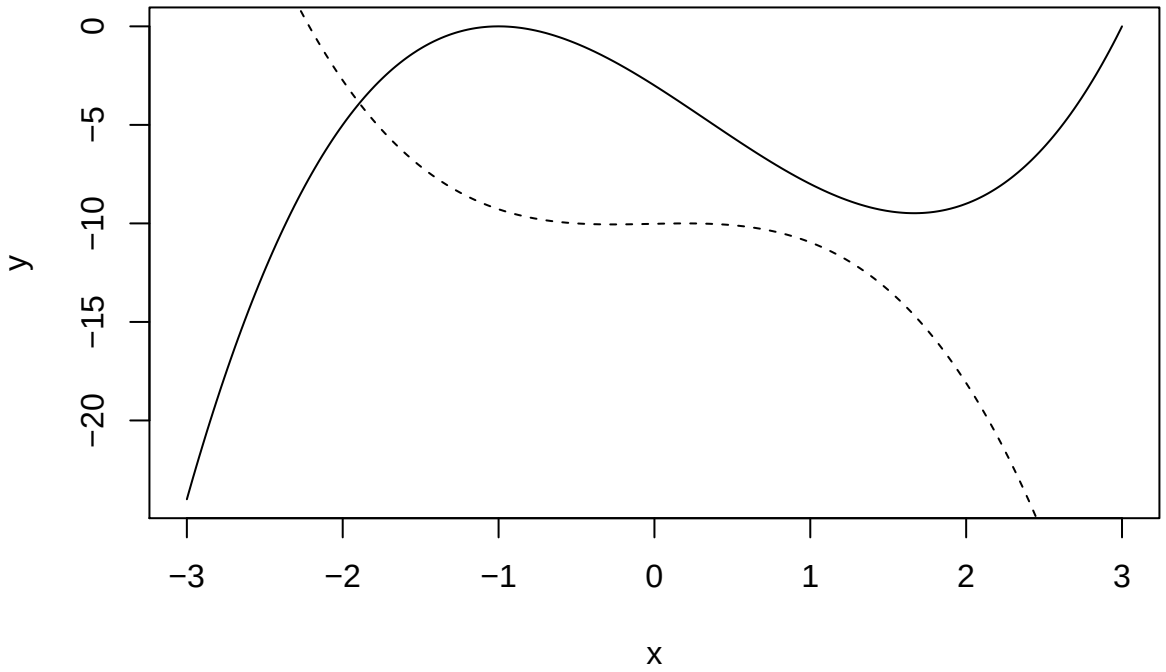
Chapter8 output

```
set.seed(76543); # recompile will have same random numbers
######



#### Creating polynomial plots
# R code for quadratic and cubic plots
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)
```




```
# R code for quadratic and cubic plots

x <- rnorm(5)
y <- x+runif(5)  #observed
x2 <- x^2
x3 <- x^3
x4 <- x^4
myfit <- lm(y ~ x + x2 + x3 + x4)
x
```

```
## [1] -0.9481572 -0.5272498  0.1235671  0.1028999 -0.3368674
```

```
y
```

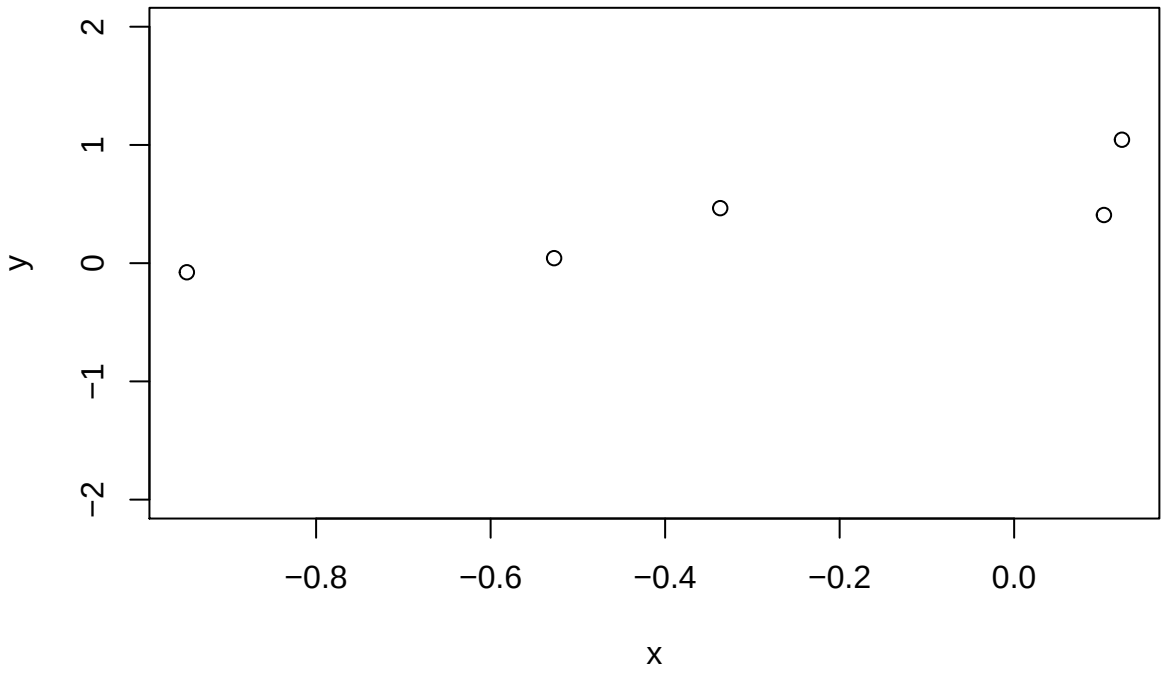
```
## [1] -0.07692973  0.04235387  1.04419157  0.40731420  0.46522970
```

```
summary(myfit)
```

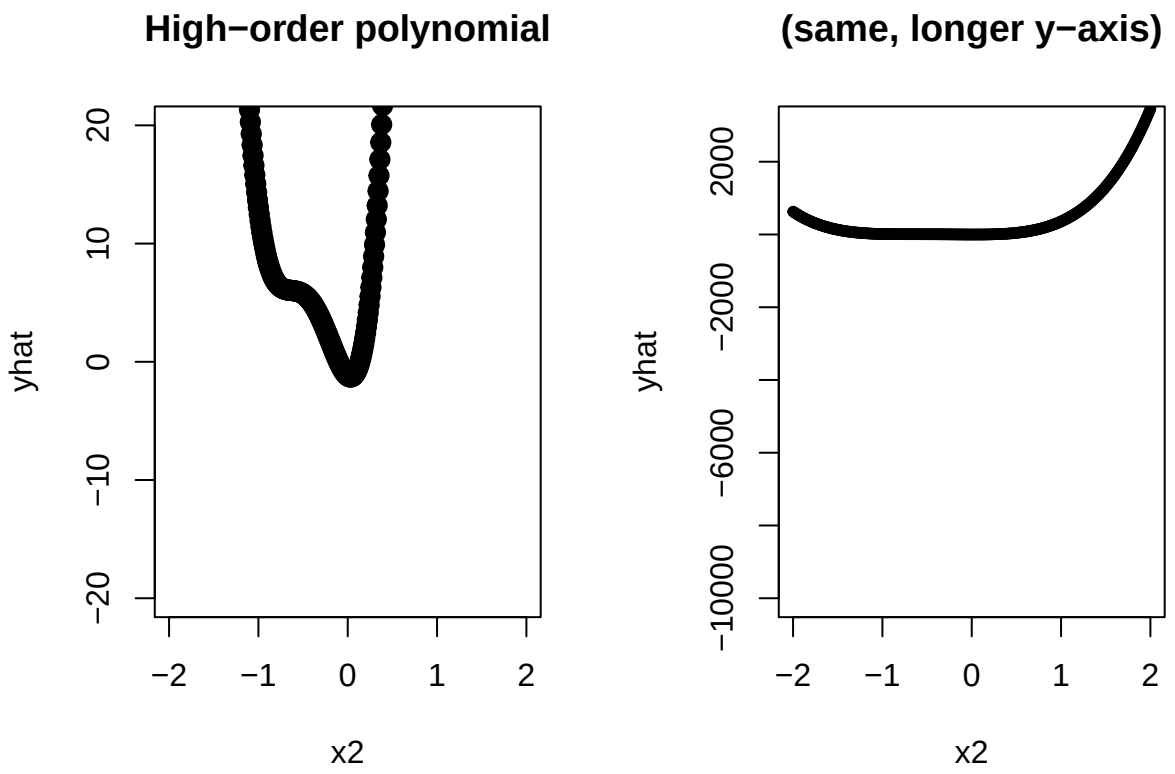
```
##
## Call:
## lm(formula = y ~ x + x2 + x3 + x4)
##
## Residuals:
## ALL 5 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.199         NA      NA       NA
## x              5.511         NA      NA       NA
## x2            78.670         NA      NA       NA
## x3           177.782         NA      NA       NA
## x4           107.847         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
```

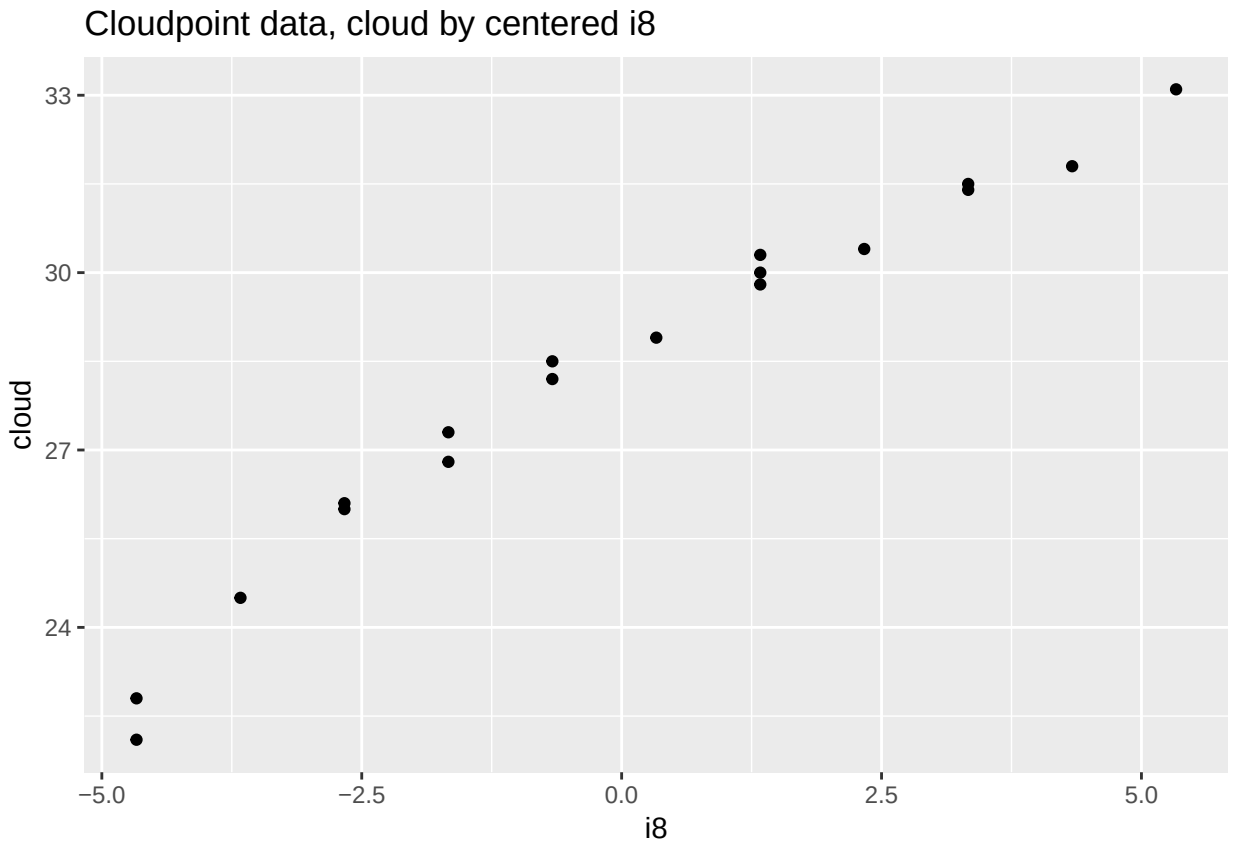
```
plot(x,y,ylim=c(-2,2))
```




```
x2 <- seq(-2,2,.01)
yhat <-myfit$coefficient[1] -myfit$coefficient[2]*x2 +myfit$coefficient[3]*x2^2 +myfit$coefficient[4]*x
par(mfrow=c(1,2))
plot(x2, yhat, main="High-order polynomial", pch=20, cex=2,ylim=c(-20,20))
points(x2, yhat, type="l", lt=1)
plot(x2,yhat, main="(same, longer y-axis)", pch=20, cex=1, ylim=c(-10000,3000))
points(x2, yhat, type="l", lt=1)


#### Example: Cloud point
cloudpoint <- read.table("http://statacumen.com/teach/ADA2/ADA2_notes_Ch08_cloudpoint.dat"
                         , header = TRUE)
# center i8 by subracting the mean
cloudpoint$i8 <- cloudpoint$i8 - mean(cloudpoint$i8)



library(ggplot2)
```




```
p <- ggplot(cloudpoint, aes(x = i8, y = cloud))
p <- p + geom_point()
p <- p + labs(title="Cloudpoint data, cloud by centered i8")
print(p)
```



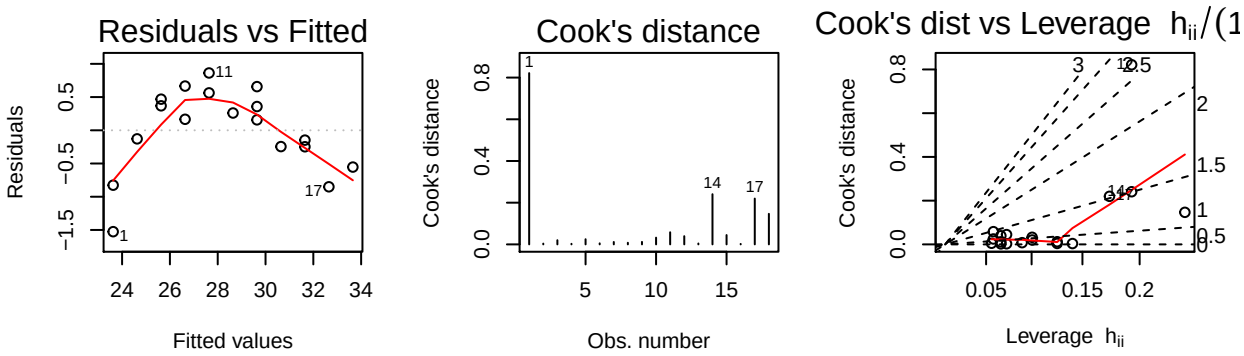

```
lm.c.i <- lm(cloud ~ i8, data = cloudpoint)
#library(car)
#Anova(aov(lm.c.i), type=3)
#summary(lm.c.i)



# plot diagnistics
par(mfrow=c(2,3))
plot(lm.c.i, which = c(1,4,6), pch=as.character(cloudpoint$type))

plot(cloudpoint$i8, lm.c.i$residuals, main="Residuals vs i8", pch=as.character(cloudpoint$type))
  # horizontal line at zero
  abline(h = 0, col = "gray75")

# Normality of Residuals
library(car)
```

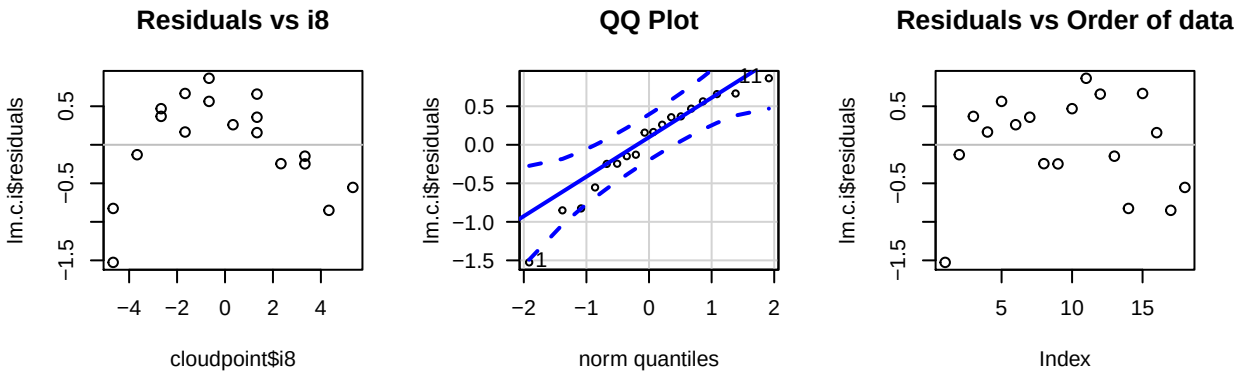
```
## Loading required package: carData
```

```
qqPlot(lm.c.i$residuals, las = 1, main="QQ Plot", pch=as.character(cloudpoint$type))
```

```
## [1]  1 11
```

```
# residuals vs order of data
plot(lm.c.i$residuals, main="Residuals vs Order of data")
  # horizontal line at zero
  abline(h = 0, col = "gray75")
```





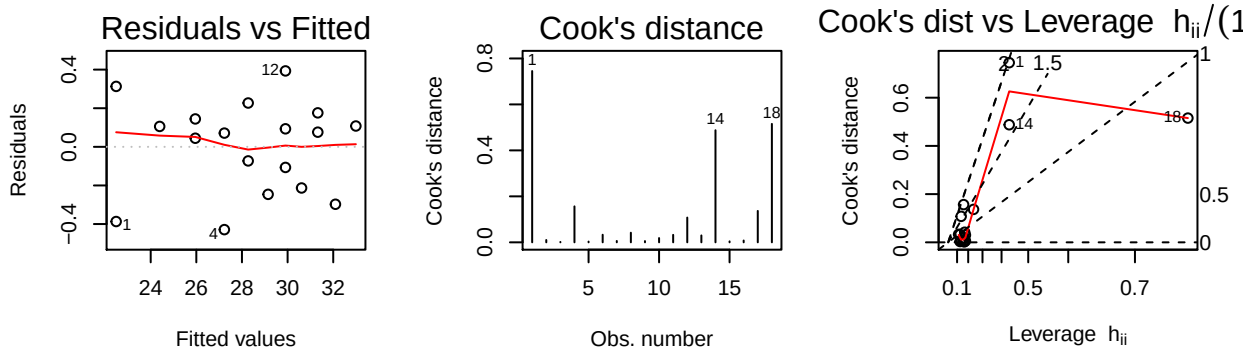

```
# I() is used to create an interpreted object treated "as is"
#   so we can include quadratic and cubic terms in the formula
#   without creating separate columns in the dataset of these terms
lm.c.i3 <- lm(cloud ~ i8 + I(i8^2) + I(i8^3), data = cloudpoint)
#library(car)
#Anova(aov(lm.c.i3), type=3)
summary(lm.c.i3)
```

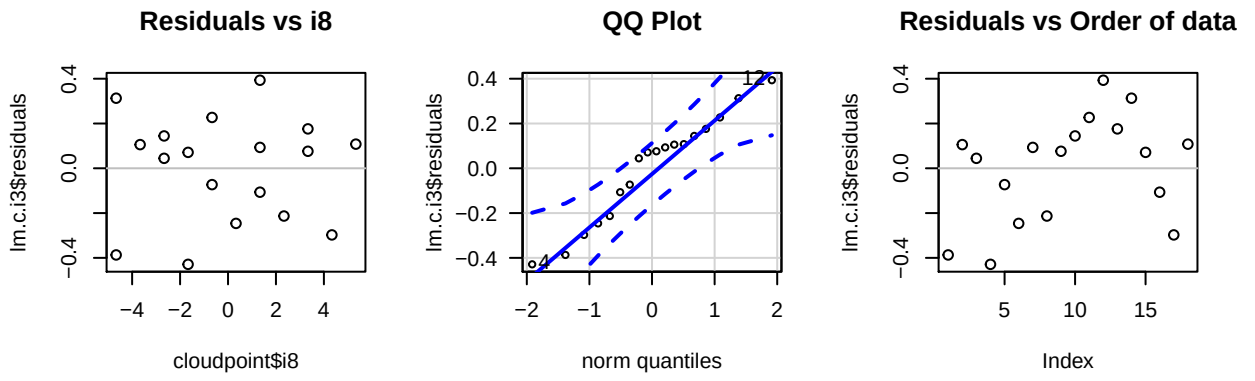
```
##
## Call:
## lm(formula = cloud ~ i8 + I(i8^2) + I(i8^3), data = cloudpoint)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42890 -0.18658  0.07355  0.13536  0.39328
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.870451   0.088364 326.723  < 2e-16 ***
## i8           0.847889   0.048536  17.469 6.67e-11 ***
## I(i8^2)     -0.065998   0.007323  -9.012 3.33e-07 ***
## I(i8^3)      0.009735   0.002588   3.762   0.0021 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2599 on 14 degrees of freedom
## Multiple R-squared:  0.9943, Adjusted R-squared:  0.9931
## F-statistic: 812.9 on 3 and 14 DF,  p-value: 6.189e-16
```

```
# plot diagnistics
par(mfrow=c(2,3))
plot(lm.c.i3, which = c(1,4,6), pch=as.character(cloudpoint$type))

plot(cloudpoint$i8, lm.c.i3$residuals, main="Residuals vs i8", pch=as.character(cloudpoint$type))
  # horizontal line at zero
  abline(h = 0, col = "gray75")

# Normality of Residuals
library(car)
qqPlot(lm.c.i3$residuals, las = 1, main="QQ Plot", pch=as.character(cloudpoint$type))
```

```
## [1]  4 12
```

```
# residuals vs order of data
plot(lm.c.i3$residuals, main="Residuals vs Order of data")
  # horizontal line at zero
  abline(h = 0, col = "gray75")
```






```
# remove points for minimum and maximum i8 values
cloudpoint2 <- cloudpoint[!(cloudpoint$i8 == min(cloudpoint$i8) |
                            cloudpoint$i8 == max(cloudpoint$i8)), ]
lm.c.i2 <- lm(cloud ~ i8 + I(i8^2) + I(i8^3), data = cloudpoint2)
#library(car)
#Anova(aov(lm.c.i2), type=3)
summary(lm.c.i2)
```

```
##
## Call:
## lm(formula = cloud ~ i8 + I(i8^2) + I(i8^3), data = cloudpoint2)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36620 -0.12845  0.03737  0.14031  0.33737
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.857039   0.089465 322.551  < 2e-16 ***
## i8           0.904515   0.058338  15.505 8.04e-09 ***
## I(i8^2)     -0.060714   0.012692  -4.784 0.000568 ***
## I(i8^3)      0.003168   0.005166   0.613 0.552200
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2313 on 11 degrees of freedom
## Multiple R-squared:  0.9917, Adjusted R-squared:  0.9894
## F-statistic: 436.3 on 3 and 11 DF,  p-value: 1.032e-11
```

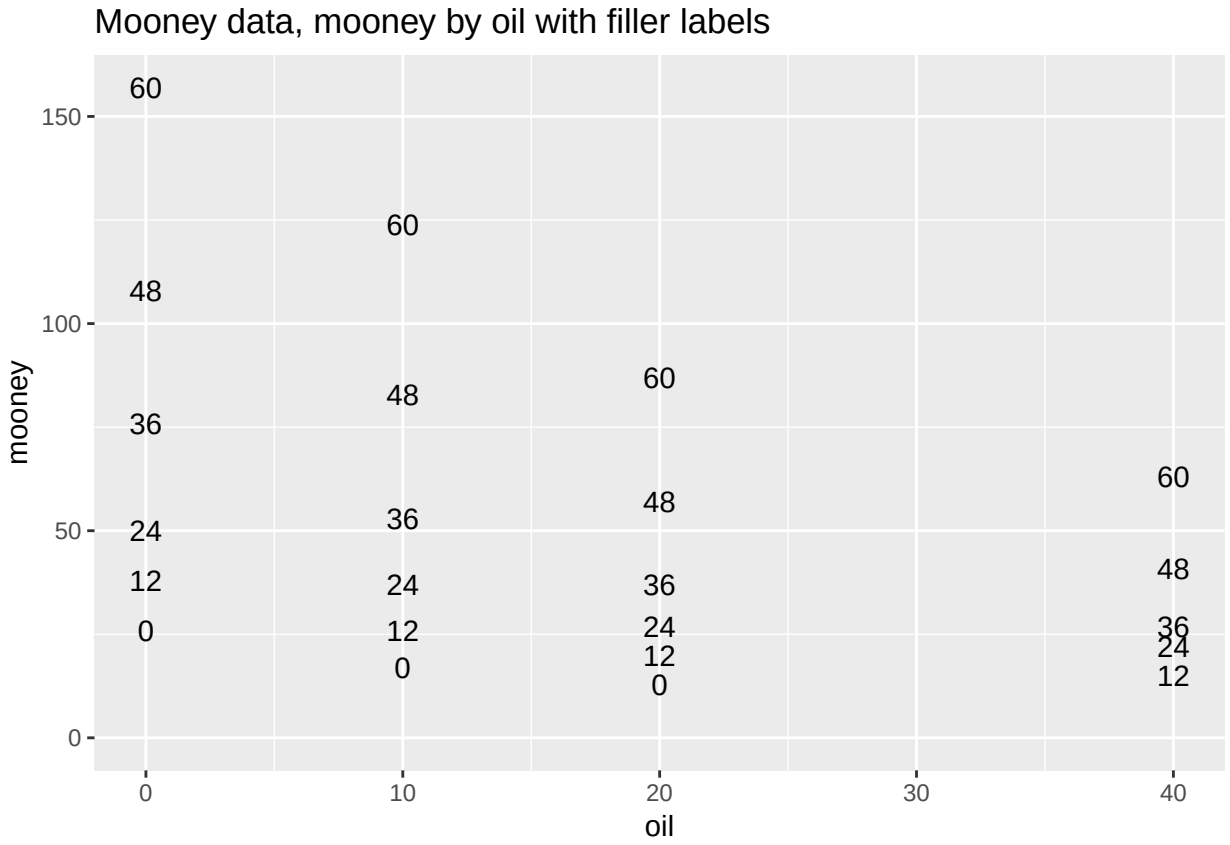
```
#### Example: Mooney viscosity
mooney <- read.table("http://statacumen.com/teach/ADA2/ADA2_notes_Ch08_mooney.dat"
                     , header = TRUE)


mooney
```

```
##    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
## 11  10     48     83
## 12  10     60    124
## 13  20      0     13
## 14  20     12     20
## 15  20     24     27
## 16  20     36     37
## 17  20     48     57
## 18  20     60     87
## 19  40      0     NA
## 20  40     12     15
## 21  40     24     22
## 22  40     36     27
## 23  40     48     41
## 24  40     60     63
```

```
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)))
```

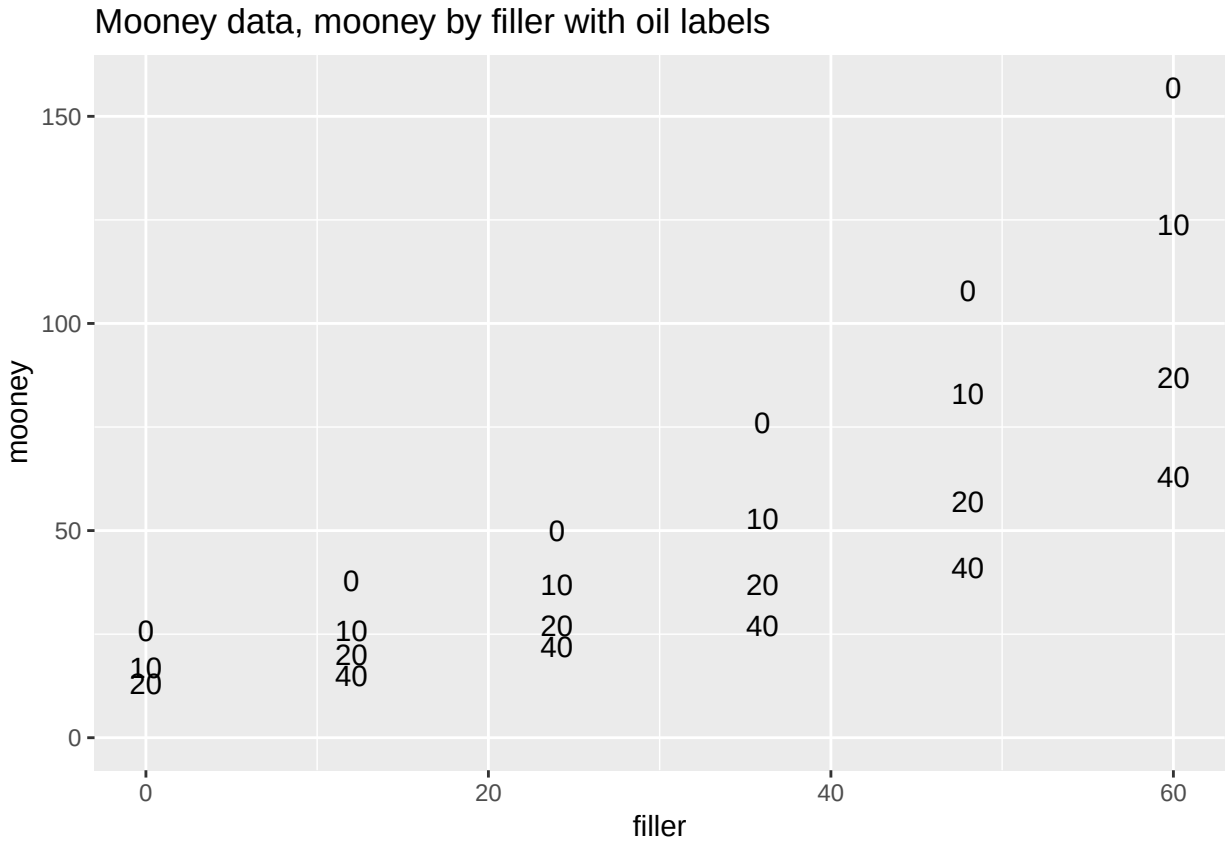
```
p <- p + labs(title="Mooney data, mooney by oil with filler labels")
print(p)
```

```
## 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()
p <- p + scale_y_continuous(limits = c(0, max(mooney$mooney, na.rm=TRUE)))
p <- p + labs(title="Mooney data, mooney by filler with oil labels")
print(p)
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```

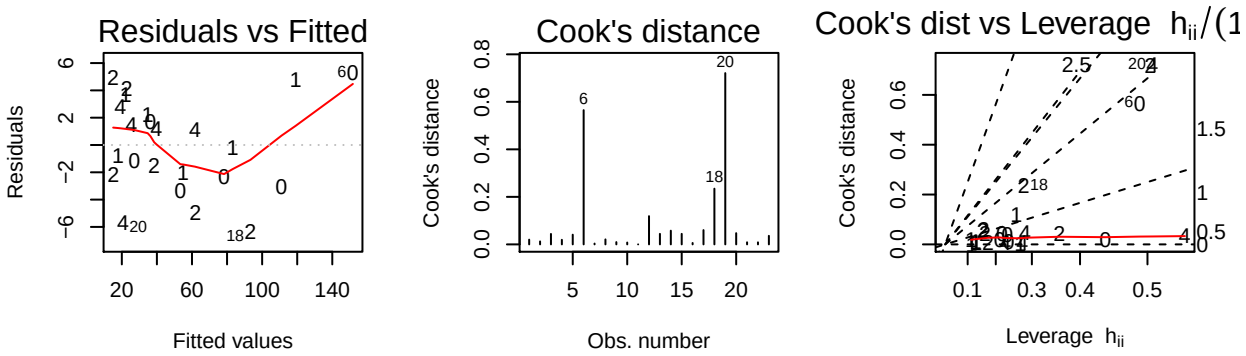


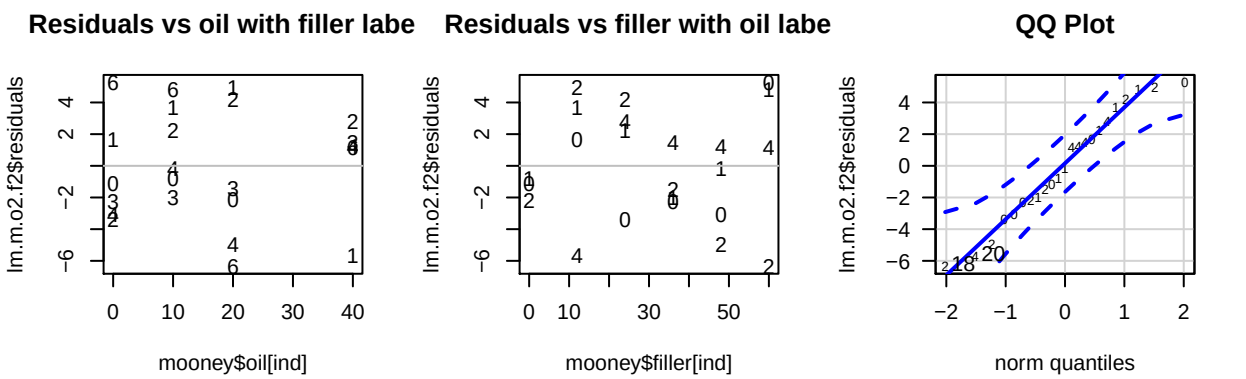

```
# I create each term separately
lm.m.o2.f2 <- lm(mooney ~ oil + filler + I(oil^2) + I(filler^2) + I(oil * filler),
                 data = mooney)
summary(lm.m.o2.f2)
```

```
##
## Call:
## lm(formula = mooney ~ oil + filler + I(oil^2) + I(filler^2) +
##     I(oil * filler), data = mooney)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3497 -2.2231 -0.1615  2.5424  5.2749
##
## 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 *
## I(oil^2)         0.033611   0.004663   7.208 1.46e-06 ***
## I(filler^2)      0.027323   0.002410  11.339 2.38e-09 ***
## I(oil * filler) -0.038659   0.003187 -12.131 8.52e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.937 on 17 degrees of freedom
##   (1 observation deleted due to missingness)
```

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

```
## poly() will evaluate variables and give joint polynomial values
##        which is helpful when you have many predictors
#head(mooney, 10)
#head(poly(mooney$oil, mooney$filler, degree = 2, raw = TRUE), 10)
## This model is equivalent to the one above
#lm.m.o2.f2 <- lm(mooney ~ poly(oil, filler, degree = 2, raw = TRUE), data = mooney)
#summary(lm.m.o2.f2)



# plot diagnistics
par(mfrow=c(2,3))
plot(lm.m.o2.f2, which = c(1,4,6), pch=as.character(mooney$oil))

# because of one missing value, get the indices of non-missing
ind <- as.numeric(names(lm.m.o2.f2$residuals))

plot(mooney$oil[ind], lm.m.o2.f2$residuals, main="Residuals vs oil with filler labels", pch=as.characte
  # horizontal line at zero
  abline(h = 0, col = "gray75")

plot(mooney$filler[ind], lm.m.o2.f2$residuals, main="Residuals vs filler with oil labels", pch=as.chara
  # horizontal line at zero
  abline(h = 0, col = "gray75")

# Normality of Residuals
library(car)
qqPlot(lm.m.o2.f2$residuals, las = 1, main="QQ Plot", pch=as.character(mooney$oil[ind]))
```

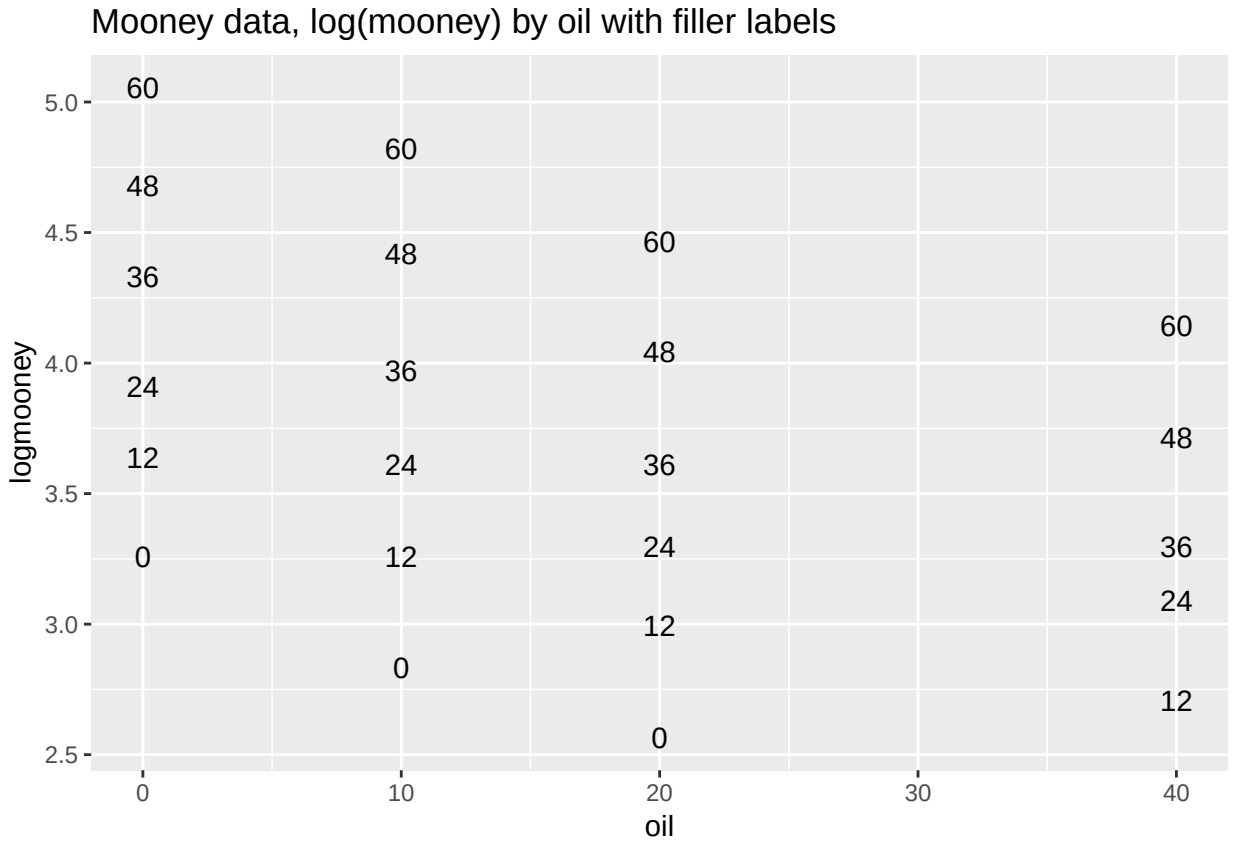





```
## 18 20
## 18 19
```

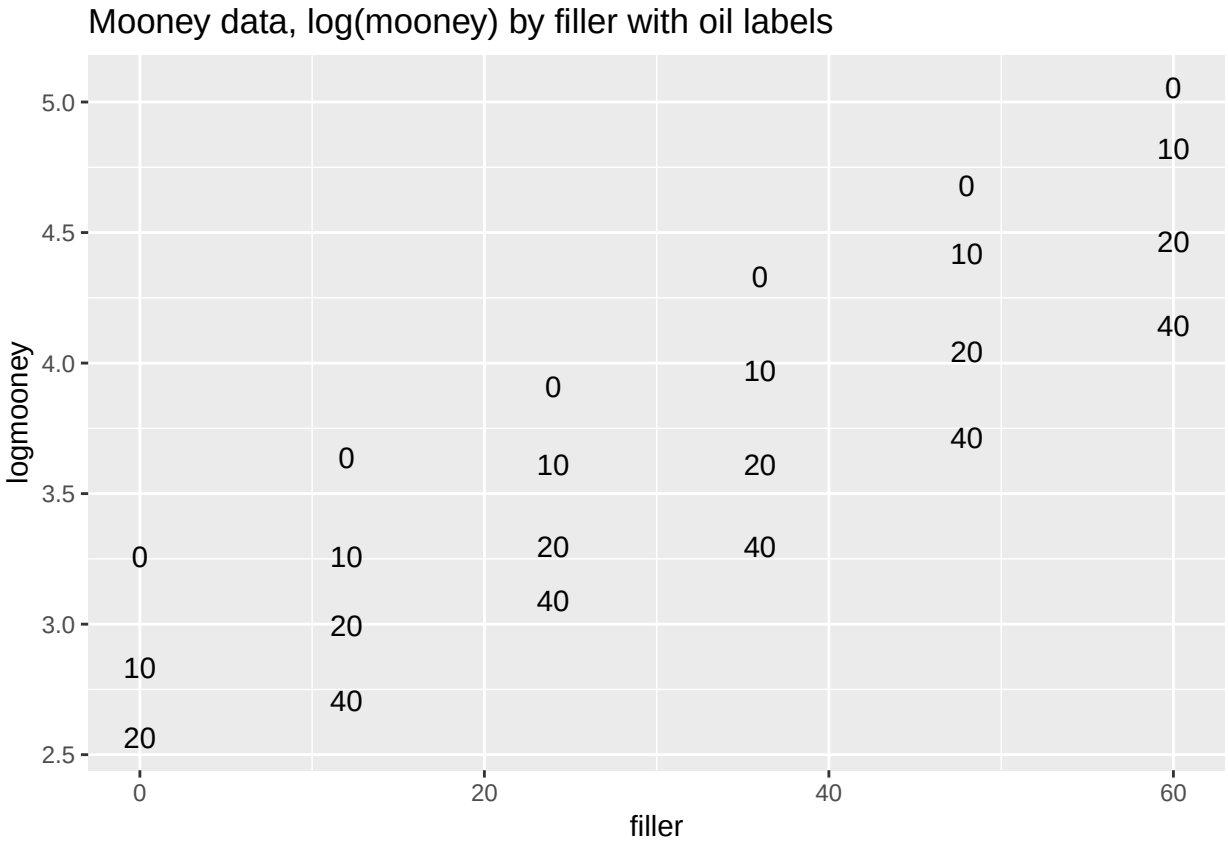
```
## residuals vs order of data
#plot(lm.m.o2.f2$residuals, main="Residuals vs Order of data")
#  # horizontal line at zero
#  abline(h = 0, col = "gray75")



# log transform the response
mooney$logmooney <- log(mooney$mooney)

library(ggplot2)
p <- ggplot(mooney, aes(x = oil, y = logmooney, label = filler))
p <- p + geom_text()
#p <- p + scale_y_continuous(limits = c(0, max(mooney$logmooney, na.rm=TRUE)))
p <- p + labs(title="Mooney data, log(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 = logmooney, label = oil))
p <- p + geom_text()
#p <- p + scale_y_continuous(limits = c(0, max(mooney$logmooney, na.rm=TRUE)))
p <- p + labs(title="Mooney data, log(mooney) by filler with oil labels")
print(p)
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```

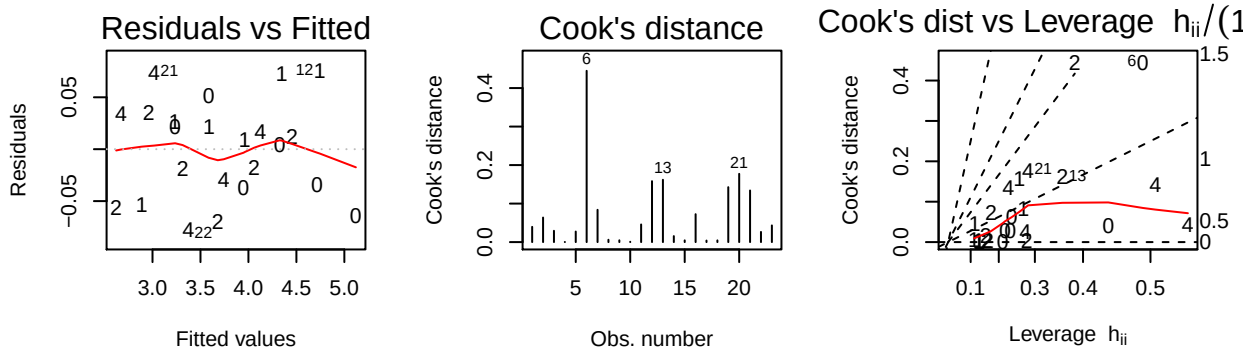


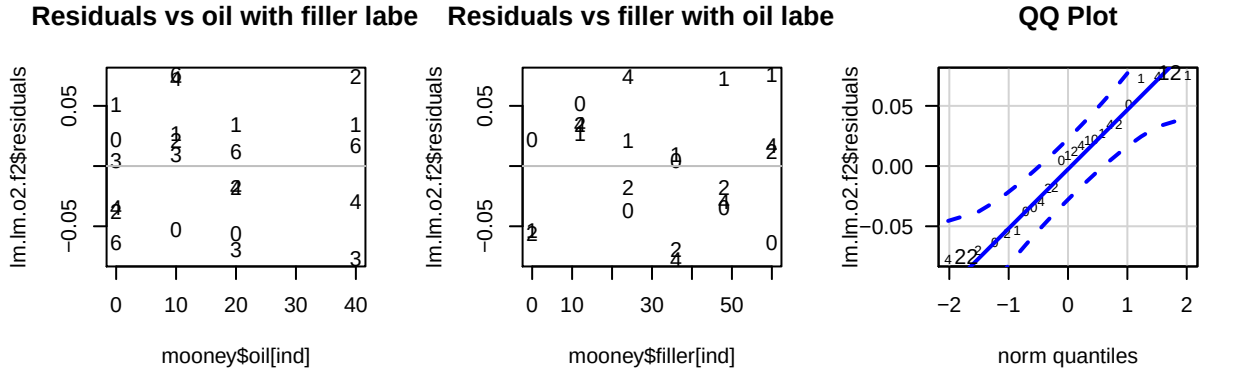

```
# I create each term separately
lm.lm.o2.f2 <- lm(logmooney ~ oil + filler + I(oil^2) + I(filler^2) + I(oil * filler),
                  data = mooney)
summary(lm.lm.o2.f2)
```

```
##
## Call:
## lm(formula = logmooney ~ oil + filler + I(oil^2) + I(filler^2) +
##     I(oil * filler), data = mooney)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.077261 -0.035795  0.009193  0.030641  0.075640
##
## 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 ***
## I(oil^2)         4.227e-04  6.339e-05   6.668 3.96e-06 ***
## I(filler^2)      4.657e-05  3.276e-05   1.421    0.173
## I(oil * filler) -4.231e-05  4.332e-05  -0.977    0.342
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05352 on 17 degrees of freedom
##   (1 observation deleted due to missingness)
```

```
## Multiple R-squared:  0.9954, Adjusted R-squared:  0.9941
## F-statistic:   737 on 5 and 17 DF,  p-value: < 2.2e-16
```

```r
# plot diagnistics
par(mfrow=c(2,3))
plot(lm.lm.o2.f2, which = c(1,4,6), pch=as.character(mooney$oil))

# because of one missing value, get the indices of non-missing
ind <- as.numeric(names(lm.lm.o2.f2$residuals))

plot(mooney$oil[ind], lm.lm.o2.f2$residuals, main="Residuals vs oil with filler labels", pch=as.charact
  # horizontal line at zero
  abline(h = 0, col = "gray75")

plot(mooney$filler[ind], lm.lm.o2.f2$residuals, main="Residuals vs filler with oil labels", pch=as.char
  # horizontal line at zero
  abline(h = 0, col = "gray75")

# Normality of Residuals
library(car)
qqPlot(lm.lm.o2.f2$residuals, las = 1, main="QQ Plot", pch=as.character(mooney$oil[ind]))
```





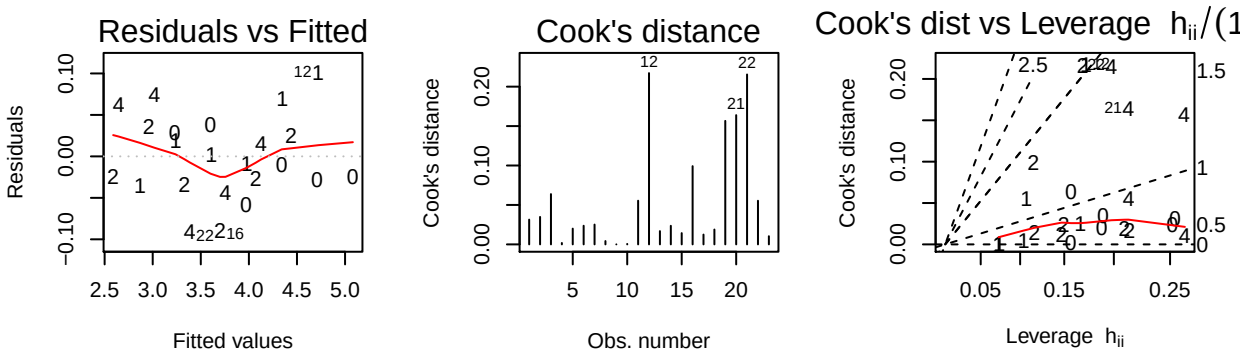

```
## 22 12
## 21 12
```

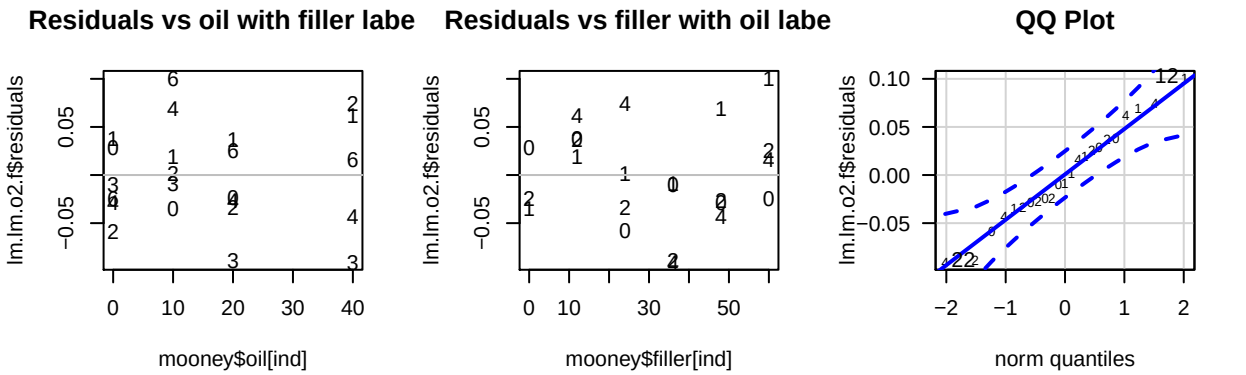
```r
## residuals vs order of data
#plot(lm.lm.o2.f2$residuals, main="Residuals vs Order of data")
#  # horizontal line at zero
#  abline(h = 0, col = "gray75")
```

```r
# I create each term separately
lm.lm.o2.f <- lm(logmooney ~ oil + filler + I(oil^2),
                    data = mooney)
summary(lm.lm.o2.f)
```

```
##
## Call:
## lm(formula = logmooney ~ oil + filler + I(oil^2), data = mooney)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.090796 -0.031113 -0.008831  0.032533  0.100587
##
## 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 ***
## I(oil^2)     4.097e-04  6.356e-05   6.446 3.53e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05423 on 19 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9947, Adjusted R-squared:  0.9939
## F-statistic:  1195 on 3 and 19 DF,  p-value: < 2.2e-16
```

```r
# plot diagnistics
par(mfrow=c(2,3))
plot(lm.lm.o2.f, which = c(1,4,6), pch=as.character(mooney$oil))

# because of one missing value, get the indices of non-missing
ind <- as.numeric(names(lm.lm.o2.f$residuals))

plot(mooney$oil[ind], lm.lm.o2.f$residuals, main="Residuals vs oil with filler labels", pch=as.characte
  # horizontal line at zero
  abline(h = 0, col = "gray75")

plot(mooney$filler[ind], lm.lm.o2.f$residuals, main="Residuals vs filler with oil labels", pch=as.chara
  # horizontal line at zero
  abline(h = 0, col = "gray75")

# Normality of Residuals
library(car)
qqPlot(lm.lm.o2.f$residuals, las = 1, main="QQ Plot", pch=as.character(mooney$oil[ind]))
```






```
## 12 22
## 12 21
```

```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/chap08plot1.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
## residuals vs order of data
#plot(lm.lm.o2.f$residuals, main="Residuals vs Order of data")
#  # horizontal line at zero
#  abline(h = 0, col = "gray75")
```