Chapter 10: An example of a regression analysis

Contents

1 Regression analysis with usual R

`lm(formula, data = ..., subset = ...)`

General form of formula: **`response ~ expression`**

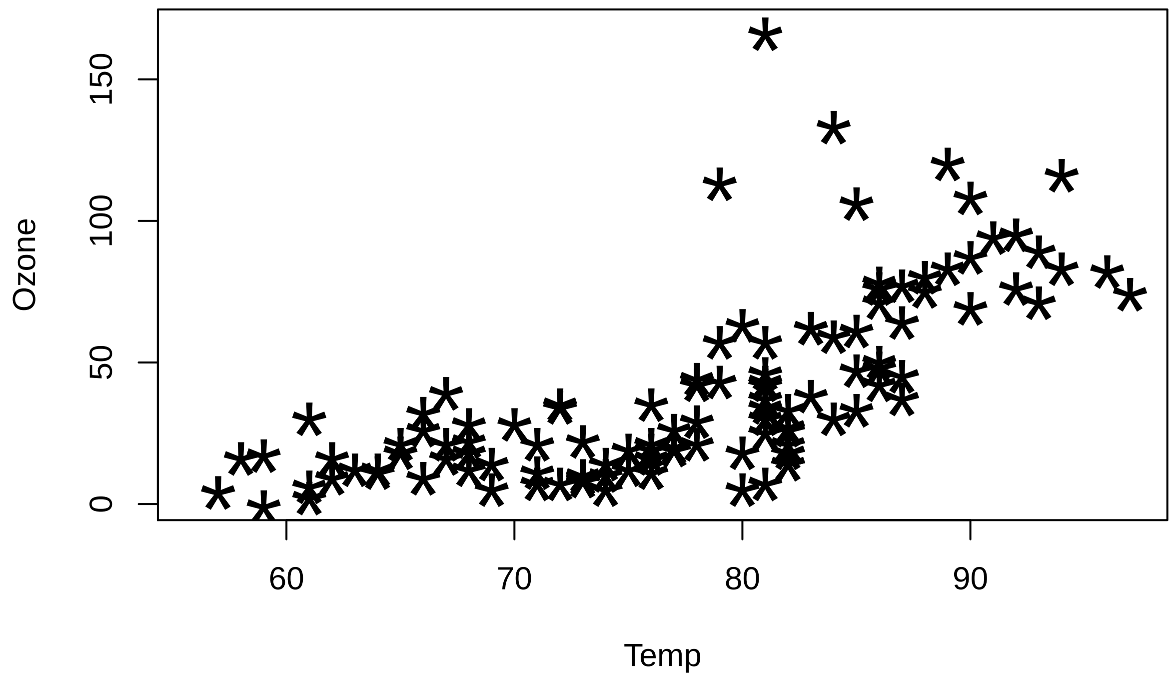
By default, the `lm()` function will print out the estimates for the coefficients. Much more is returned, but needs to be asked for.
Some useful extractors are:

- `summary()`
- `plot()`
- `coef()`
- `residuals()`
- `fitted()`
- `deviance()`
- `predict()`
- `anova()`
- `AIC()`

Example *chol*
We use the data set `airquality` from the package `datasets`.

We first make a scatterplot of `Ozone` versus `Temp`

```
plot(Ozone ~ Temp, data = airquality, pch = "*", cex = 3)
```



We perform a linear regression analysis (`lm(response ~ X)`).

```r
res.lm <- lm(Ozone ~ Temp, data = airquality)
summary(res.lm)
```

```
##
## Call:
## lm(formula = Ozone ~ Temp, data = airquality)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.729 -17.409  -0.587  11.306 118.271
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -146.9955    18.2872  -8.038 9.37e-13 ***
## Temp           2.4287     0.2331  10.418  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.71 on 114 degrees of freedom
##   (37 observations deleted due to missingness)
## Multiple R-squared:  0.4877, Adjusted R-squared:  0.4832
## F-statistic: 108.5 on 1 and 114 DF,  p-value: < 2.2e-16
```

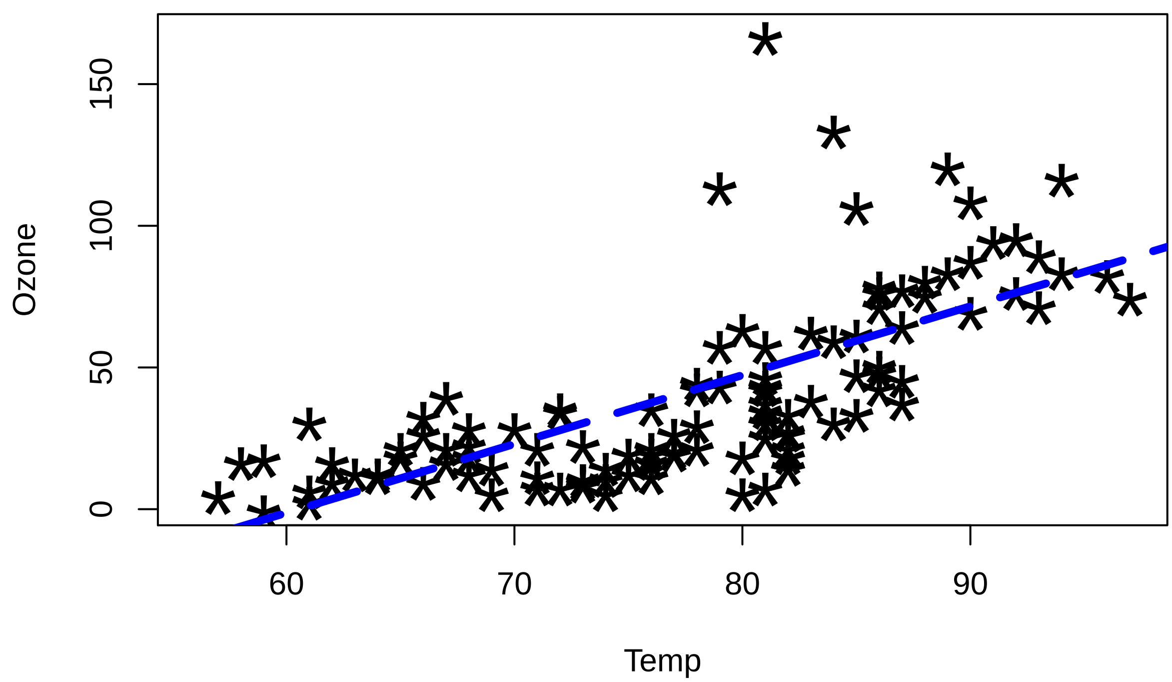
From the output:
Ozone = -147 + 2.43 Temp

```r
names(res.lm)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "na.action"     "xlevels"       "call"          "terms"
## [13] "model"
```

We add the regression line to the plot

```r
plot(Ozone ~ Temp, data = airquality, pch = "*", cex = 3)
abline(res.lm, lty = 5, col = 4, lwd = 4)
```



The argument **formula** in the function `lm` can take more complex forms than the example above. Some examples are

Formula regression model	Formula argument in R
$Z = \beta_0 + \beta_1 X + \beta_2 Y$	`Z ~ X + Y`
$Z = \beta_0 + \beta_1 X + \beta_2 X^2$	`Z ~ X + I(X^2)`
$Z = \beta_1 X$	`Z ~ X -1`

Note: Function `I()` is a conversion of object function. It means that the object X^2 should be treated as a new variable.
It can be done in another way: First, create a new variable $X_2 = X \cdot X$. Then use this variable in the regression model.

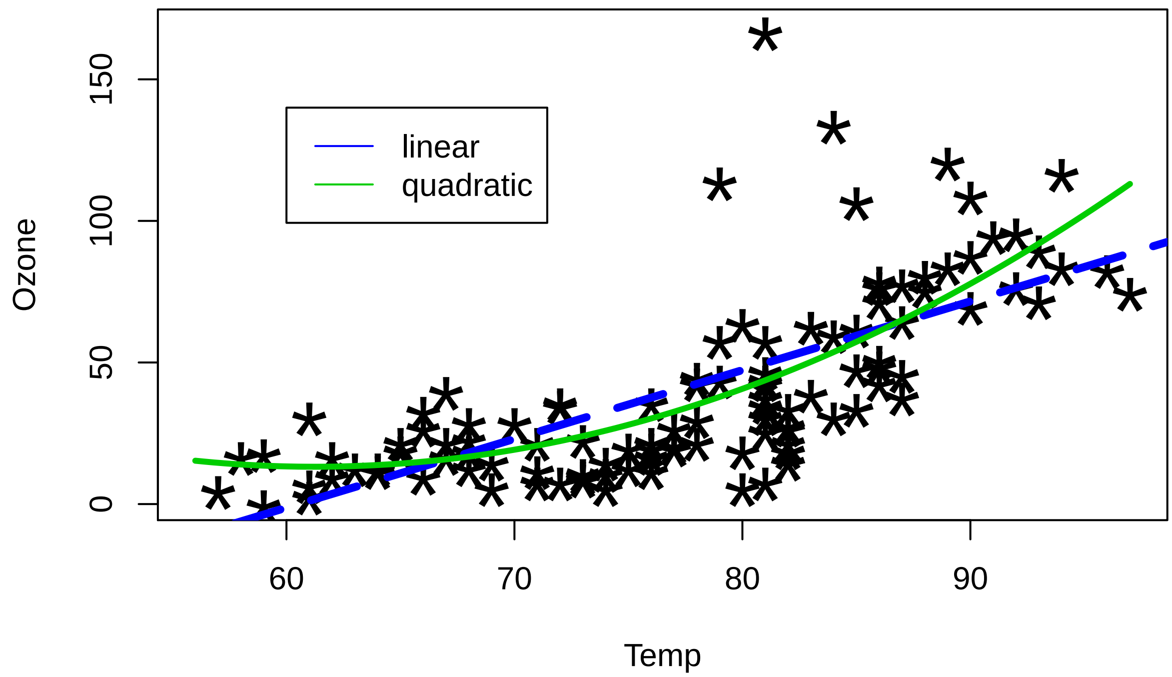
We will also try to apply a polynomial model to the data.

```r
res.lm2 <- lm(Ozone ~ Temp + I(Temp^2), data = airquality)
summary(res.lm2)
```

```
##
## Call:
## lm(formula = Ozone ~ Temp + I(Temp^2), data = airquality)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.619 -12.513  -2.736   9.676 123.909
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 305.48577  122.12182   2.501 0.013800 *
## Temp         -9.55060    3.20805  -2.977 0.003561 **
## I(Temp^2)     0.07807    0.02086   3.743 0.000288 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.47 on 113 degrees of freedom
##   (37 observations deleted due to missingness)
## Multiple R-squared:  0.5442, Adjusted R-squared:  0.5362
## F-statistic: 67.46 on 2 and 113 DF,  p-value: < 2.2e-16
```

From the output:
Ozone = 305.49 - 9.55 Temp + 0.078 Temp^2

```r
# One possibility
plot(Ozone ~ Temp, data = airquality, pch = "*", cex = 3)

# Add regression line
abline(res.lm, lty = 5, col = 4, lwd = 4)

# Add quadratic curve
curve(305.5 - 9.55*x + 0.078*x*x, add = T, col = 3, lwd = 3)

legend(60, 140, legend = c("linear", "quadratic"), lty = c(1,1), col = c(4,3))
```



Another possibility:

- *Step 1:*
  Compute the formula for polynomial regression

```
poly <- function(x, coefs)
{
  tot <- 0
  for (i in 1:length(coefs))
  {tot <- tot + coefs[i]*x^{i-1}}
  tot
}
```

```
coef(res.lm2)
```

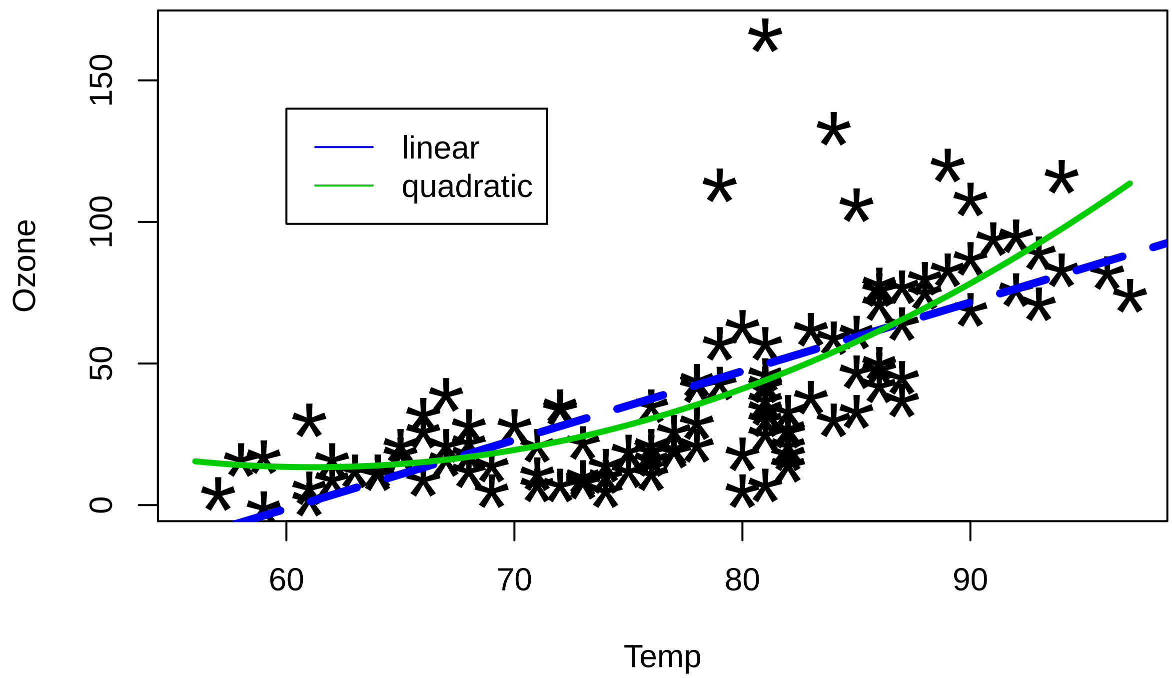
```
##   (Intercept)          Temp     I(Temp^2)
## 305.48576778  -9.55060391   0.07806798
```

What is happening in the `for` loop of `poly` for the polynomial regression model?

$\text{i} = 1 \rightarrow$	$\texttt{tot} = 0 + \text{coef}[1] \cdot 1$	$\rightarrow \texttt{tot} = 305$
$\text{i} = 2 \rightarrow$	$\texttt{tot} = 305 + \text{coef}[2] \cdot x$	$\rightarrow \texttt{tot} = 305 - 9.55 \cdot x$
$\text{i} = 3 \rightarrow$	$\texttt{tot} = 305 - 9.55 \cdot x + \text{coef}[3] \cdot x^2$	$\rightarrow \texttt{tot}$ $= 305 - 9.55 \cdot x + 0.078 \cdot x^2$

- *Step 2:*
  Produce the plot

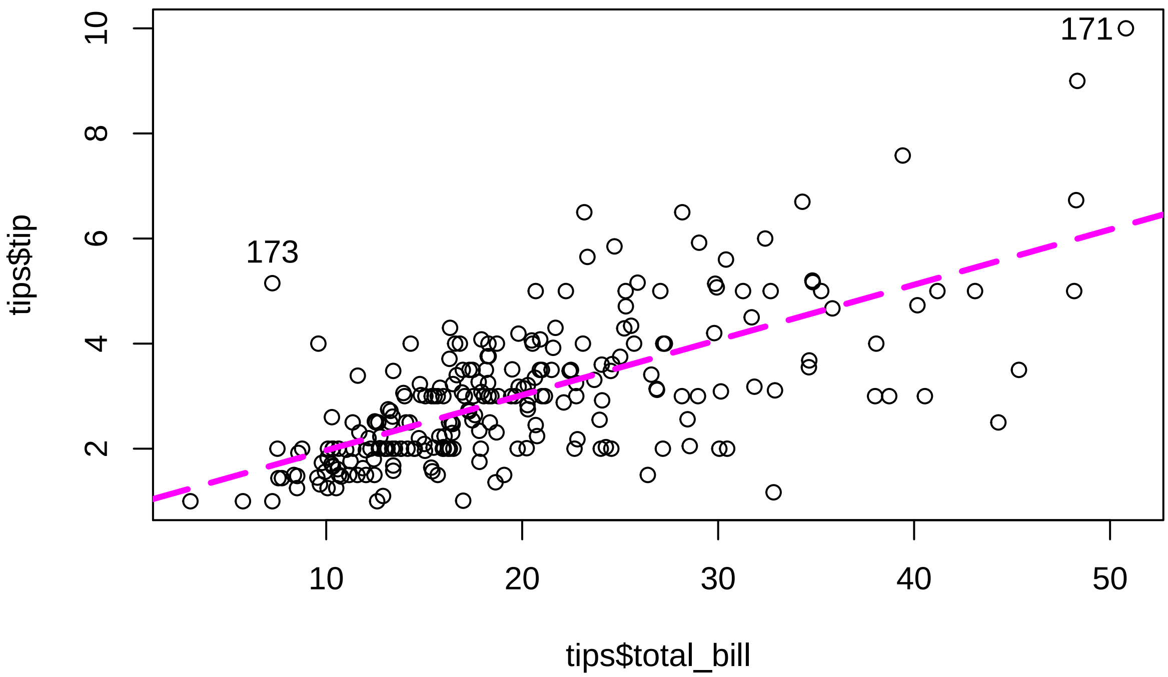
```r
# One possibility
plot(Ozone ~ Temp, data = airquality, pch = "*", cex = 3)

# Add regression line
abline(res.lm, lty = 5, col = 4, lwd = 4)

curve(poly(x, coef(res.lm2)), add = TRUE, col = 3, lwd = 3)

legend(60, 140, legend = c('linear', 'quadratic'), lty = c(1,1), col = c(4,3))
```



Remark:
The function `curve()` has an option to specify the function that should be plotted. In our case it is a function `poly(x, coef(res.lm2))`. In general, it can be any other function: `sin`, `cos`, `tan`

2 Exercises

Used data: `tips` from `reshape` package
Try to set up a regression model to predict the `tip` by `total_bill`. Add this regression line to the plot. Identify some special observations.



171
173
10
8
6
4
2
tips$tip
10
20
30
40
50
tips$total_bill