Autor: Andrés García Medina

email: andres.garcia.medina@uabc.edu.mx

In [1]:
```
data(cars)
```

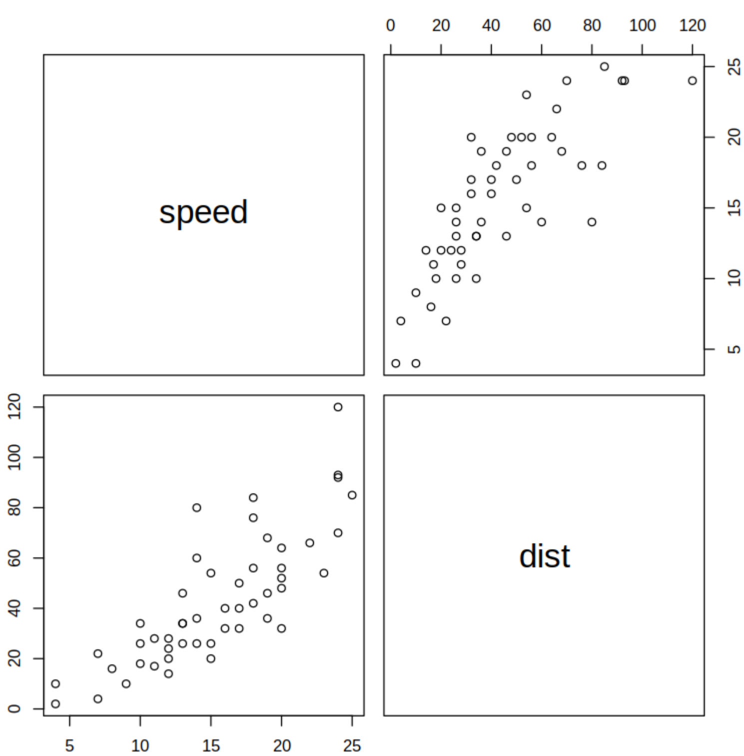
In [2]:
```
print(dim(cars))
```

```
[1] 50  2
```

In [3]:
```
print(cars[1:10,])
```

```
   speed dist
1      4    2
2      4   10
3      7    4
4      7   22
5      8   16
6      9   10
7     10   18
8     10   26
9     10   34
10    11   17
```

In [4]:
```
pairs(cars)
```



In [5]:
```
cm1 <- lm(dist ~ speed + I(speed^2), cars)
```

In [6]:
```
summary(cm1)
```

```
Call:
lm(formula = dist ~ speed + I(speed^2), data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-28.720  -9.184  -3.188   4.628  45.152 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.47014   14.81716   0.167    0.868
speed        0.91329    2.03422   0.449    0.656
I(speed^2)   0.09996    0.06597   1.515    0.136

Residual standard error: 15.18 on 47 degrees of freedom
Multiple R-squared:  0.6673,	Adjusted R-squared:  0.6532 
F-statistic: 47.14 on 2 and 47 DF,  p-value: 5.852e-12
```

La tercera columna nos da el parámetro estimado dividido por su error estándar estimado:

bajo $H_0 : \beta_j = 0$,

$T_j \sim t_{n-p}$

Residual estandar error:

$\hat{\sigma}^2 = \sum \hat{e}_i^2/(n-p)$

donde $n-p$ son los grados de libertad

Multiple R-squared:

$r^2 = 1 - \frac{\sum \hat{e}_i^2/n}{\sum(y_i - \bar{y})^2/n}$

Adjusted R-squared:

$r^2_{adj} = 1 - \frac{\sum \hat{e}_i^2/(n-p)}{\sum(y_i - \bar{y})^2/(n-1)}$

F-statistics:

Mide a hipótesis nula de que los datos fueron generados solamente por el intercepto contra la alternativa de que fueron generados por el modelo completo

Es posible extraer los componentes del modelo

In [7]:
```
cm1$coefficients
```

(Intercept): 2.47013778506624 **speed:** 0.91328761424259 **I(speed^2):** 0.0999593020698438

In [8]:
```
cm1$df.residual
```

47

In [9]:
```
cm1$residuals
```

1: -5.72263707515412 **2:** 2.27736292484591 **3:** -9.76115688618673 **4:** 8.23884311381328 **5:** -0.173834031476965 **6:** -8.7864297809069 **7:** -3.59894413447652 **8:** 4.40105586552348 **9:** 12.4010558655235 **10:** -7.61137709218583 **11:** 3.38862290781417 **12:** -13.8237286540348 **13:** -7.82372865403483 **14:** -3.82372865403483 **15:** 0.176271345965173 **16:** -5.23599882002351 **17:** 2.76400117997649 **18:** 2.76400117997649 **19:** 14.7640011799765 **20:** -8.84818759015188 **21:** 1.15181240984812 **22:** 25.1518124098481 **23:** 45.1518124098481 **24:** -18.6602949644199 **25:** -12.6602949644199 **26:** 15.3397050355801 **27:** -10.6723209428277 **28:** -2.67232094282769 **29:** -14.8842655253751 **30:** -6.88426552537512 **31:** 3.11573447462488 **32:** -9.29612871206225 **33:** 4.70387128793775 **34:** 24.7038712879377 **35:** 32.7038712879378 **36:** -19.9079105028891 **37:** -9.90791050288906 **38:** 12.0920894971109 **39:** -28.7196108978556 **40:** -12.7196108978556 **41:** -8.71961089785556 **42:** -4.71961089785556 **43:** 3.28038910214444 **44:** -4.94276750020761 **45:** -22.3542237075932 **46:** -11.9655985191184 **47:** 10.0344014808816 **48:** 11.0344014808816 **49:** 38.0344014808816 **50:** -2.77689193478336

In [10]:
```
summary(cm1)$fstatistic
```

value: 47.1407481288433 **numdf:** 2 **dendf:** 47

In [11]:
```
summary(cm1)$adj.r.squared
```

0.65317468105924

In [12]:
```
summary(cm1)$r.squared
```

0.66733081652621

In [13]:
```
cm1$model[1:10,]
```

A data.frame: 10 × 3

	dist	speed	I(speed^2)
	<dbl>	<dbl>	<I<dbl>>
1	2	4	16
2	10	4	16
3	4	7	49
4	22	7	49
5	16	8	64
6	10	9	81
7	18	10	100
8	26	10	100
9	34	10	100
10	17	11	121

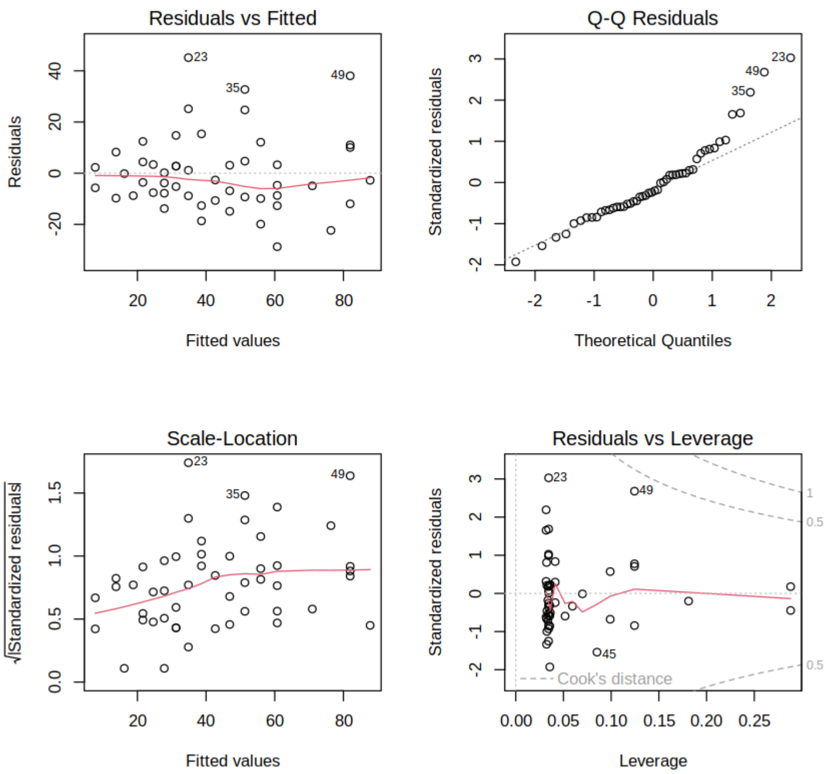
In [14]:
```
model.matrix(cm1)[1:10,]
```

A matrix: 10 × 3 of type dbl

	(Intercept)	speed	I(speed^2)
1	1	4	16
2	1	4	16
3	1	7	49
4	1	7	49
5	1	8	64
6	1	9	81
7	1	10	100
8	1	10	100
9	1	10	100
10	1	11	121

Una vez ajustado el modelo es importante revisar la plausibilidad de las supocisiones de manera gráfica

In [15]:
```
par(mfrow=c(2,2))
plot(cm1)
```



Explicación:

- Residuals vs. Fitted:
  - $\hat{e}_i = y - \hat{\mu}$ vs. $\hat{\mu} = X\hat{\beta}$
  - No se observa ningún problema evidente.
- Scale-Location:
  - Los residuales se estandarizan dividiendo por $\hat{\sigma}\sqrt{1 - A_{ii}}$, donde A es la matriz de influencia.
  - La raiz cuadrada reduce la asimetría de la distribución
  - No se observa ningún problema evidente
- Normal Q-Q:
  - los residuales estandarizados se ordenan y grafican respecto los cuantiles de la distribución normal estándar
  - la supocisión de normalidad parece plausible.
- Residuales vs. leverage:

- Los leverage A_{ii} miden el efecto potencial de que una observación particular influya en el ajuste del modelo globalmente.

- La combinación de valores altos de residuales y leverage implican que la observación tiene un efecto substancial en el ajuste del modelo.

- La distancia de Cook mide la influencia de cada observación o dato en el ajuste del modelo:

  $d_k = \frac{1}{(p+1)\hat{\sigma}^2} \sum_{i=1}^{n} (\hat{\mu}_i^k - \hat{\mu}_i)^2$

  donde el superindice k se refiere a que se ha omitido el dato k en el ajuste, por lo que un valor alto de d_k implica que el punto k influye significaticvamente en el modelo.

- En esta figura ningún dato se aleja demasiado

Nota:

- Los residuales han sido estandarizados de tal manera que si las suposiciones se satisfacen entonces se deben comportar como desviaciones del tipo N(0,1)
- Por default los 3 valores más extremos se resaltan
- Ajustemos un modelo sin los registros 23,35, 49

Consideremos otras propuestas

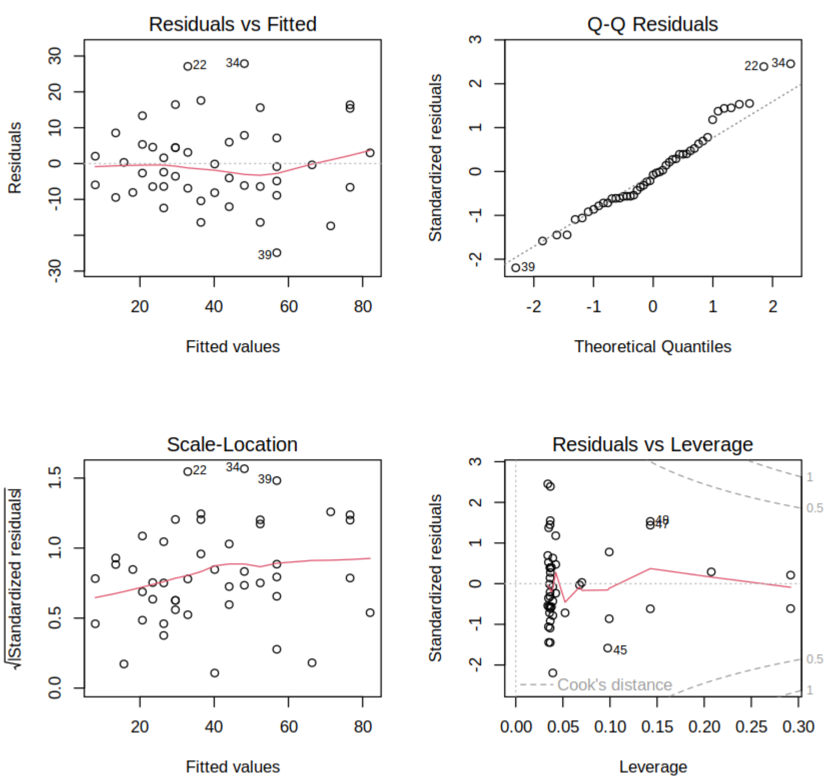
In [16]:
```
cm1b <- lm(dist ~ speed + I(speed^2), cars[-c(23,35,49),])
cm2 <- lm(dist ~ speed + I(speed^2)-1, cars)
cm2b <- lm(dist ~ speed + I(speed^2)-1, cars[-c(23,35,49),])
cm3 <- lm(dist ~ I(speed^2), cars)
cm3b <- lm(dist ~ I(speed^2), cars[-c(23,35,49),])
```

In [17]:
```
summary(cm1b)
```

```
Call:
lm(formula = dist ~ speed + I(speed^2), data = cars[-c(23, 35,
    49), ])

Residuals:
     Min       1Q   Median       3Q      Max
-24.8686  -6.7502  -0.8686   5.6539  27.8759

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.23123   11.44257   0.282   0.7790
speed        0.80363    1.59112   0.505   0.6160
I(speed^2)   0.09391    0.05217   1.800   0.0787 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.55 on 44 degrees of freedom
Multiple R-squared:  0.7451,	Adjusted R-squared:  0.7335
F-statistic:  64.3 on 2 and 44 DF,  p-value: 8.731e-14
```

In [18]:

```r
par(mfrow=c(2,2))
plot(cm1b)
```

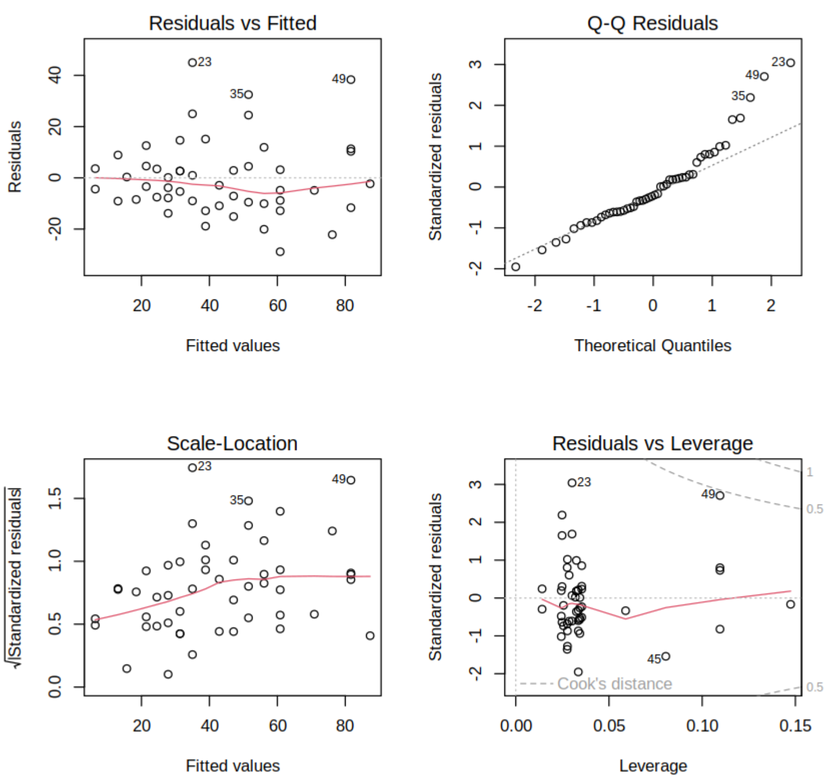


In [19]:

```r
summary(cm2)
```

```
Call:
lm(formula = dist ~ speed + I(speed^2) - 1, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-28.836  -9.071  -3.152   4.570  44.986 

Coefficients:
           Estimate Std. Error t value Pr(>|t|)   
speed       1.23903    0.55997   2.213  0.03171 * 
I(speed^2)  0.09014    0.02939   3.067  0.00355 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.02 on 48 degrees of freedom
Multiple R-squared:  0.9133,	Adjusted R-squared:  0.9097 
F-statistic: 252.8 on 2 and 48 DF,  p-value: < 2.2e-16
```

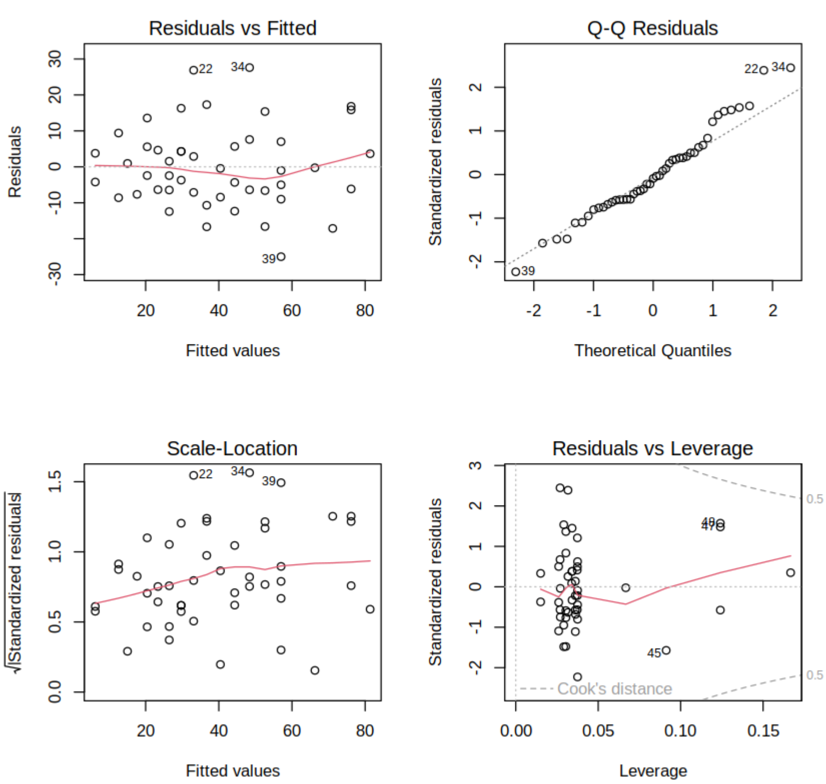
In [20]:
```
par(mfrow=c(2,2))
plot(cm2)
```



In [21]:
```
summary(cm2b)
```

```
Call:
lm(formula = dist ~ speed + I(speed^2) - 1, data = cars[-c(23,
    35, 49), ])

Residuals:
    Min      1Q  Median      3Q     Max
-25.009  -6.872  -1.009   5.618  27.600

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
speed       1.23494    0.44127   2.799  0.00753 **
I(speed^2)  0.08077    0.02337   3.457  0.00120 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.44 on 45 degrees of freedom
Multiple R-squared:  0.9394,    Adjusted R-squared:  0.9367
F-statistic: 348.6 on 2 and 45 DF,  p-value: < 2.2e-16
```

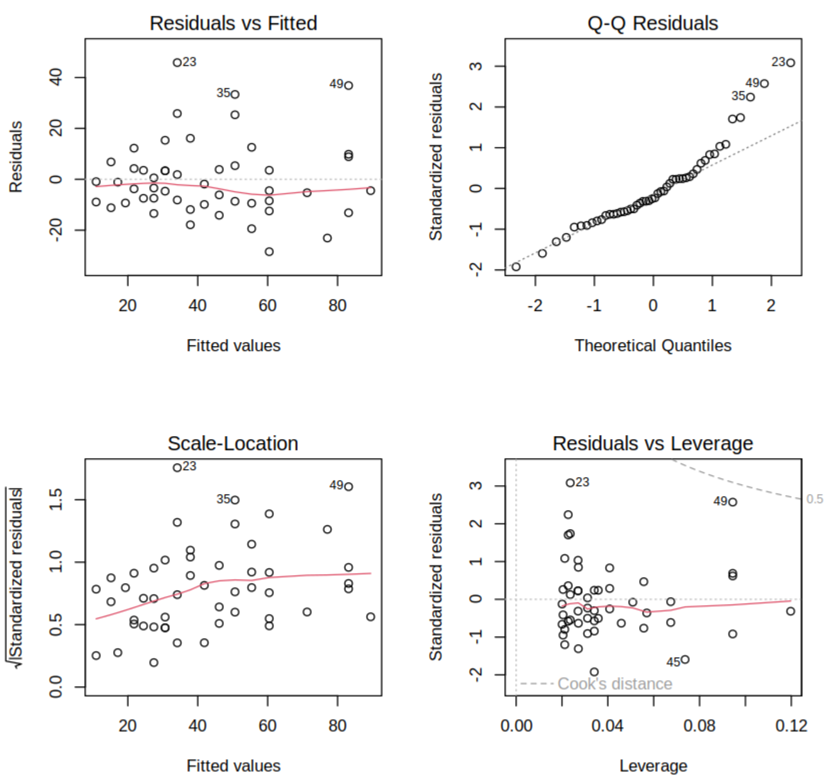
In [22]:
```
par(mfrow=c(2,2))
plot(cm2b)
```



In [23]:
```
summary(cm3)
```

```
Call:
lm(formula = dist ~ I(speed^2), data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-28.448  -9.211  -3.594   5.076  45.862 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  8.86005    4.08633   2.168   0.0351 *  
I(speed^2)   0.12897    0.01319   9.781  5.2e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.05 on 48 degrees of freedom
Multiple R-squared:  0.6659,	Adjusted R-squared:  0.6589 
F-statistic: 95.67 on 1 and 48 DF,  p-value: 5.2e-13
```

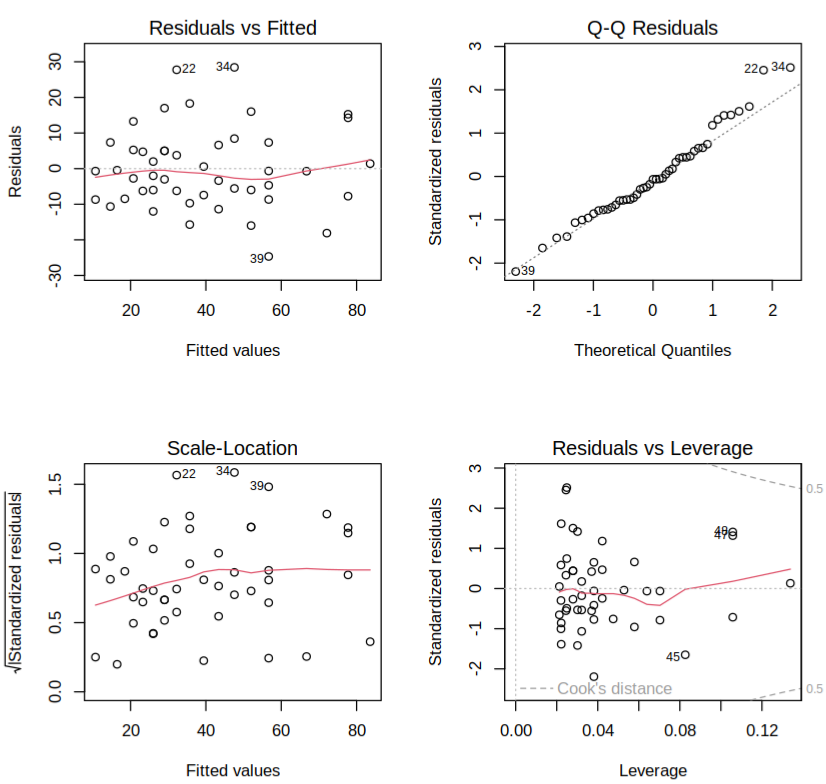
In [24]:
```
par(mfrow=c(2,2))
plot(cm3)
```



In [25]:
```
summary(cm3b)
```

```
Call:
lm(formula = dist ~ I(speed^2), data = cars[-c(23, 35, 49), ])

Residuals:
     Min       1Q   Median       3Q      Max 
-24.6651  -7.5806  -0.7212   5.9364  28.4332 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  8.77901    3.17971   2.761  0.00831 ** 
I(speed^2)   0.11972    0.01048  11.424 6.84e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.46 on 45 degrees of freedom
Multiple R-squared:  0.7436,	Adjusted R-squared:  0.7379 
F-statistic: 130.5 on 1 and 45 DF,  p-value: 6.837e-15
```

In [26]:
```
par(mfrow=c(2,2))
plot(cm3b)
```



Seleccion de Modelo

In [27]:
```
AIC(cm1,cm1b,cm2,cm2b,cm3,cm3b)
```

```
Warning message in AIC.default(cm1, cm1b, cm2, cm2b, cm3, cm3b):
“models are not all fitted to the same number of observations”
```

A data.frame: 6 × 2

	df	AIC
	<dbl>	<dbl>
cm1	4	418.7721
cm1b	4	368.3032
cm2	3	416.8016
cm2b	3	366.3883
cm3	3	416.9860
cm3b	3	366.5749

Nos quedamos con el modelo cm2b, es decir, sin intercepto y datos atipicos.

Estimamos tiempo de frenado con el mejor modelo

In [28]:
```
b <- coef(cm2b)
b
```

speed: 1.23494223751678 **I(speed^2):** 0.0807745330928482

Distancia proporcional a la velocidad

$d \sim v$

$d = \beta v$

antes de comenzar a parar

$\text{tiempo} = \text{distancia/velocidad} = \frac{\beta v}{v} = \beta$

finalmente se realiza la conversion de unidades

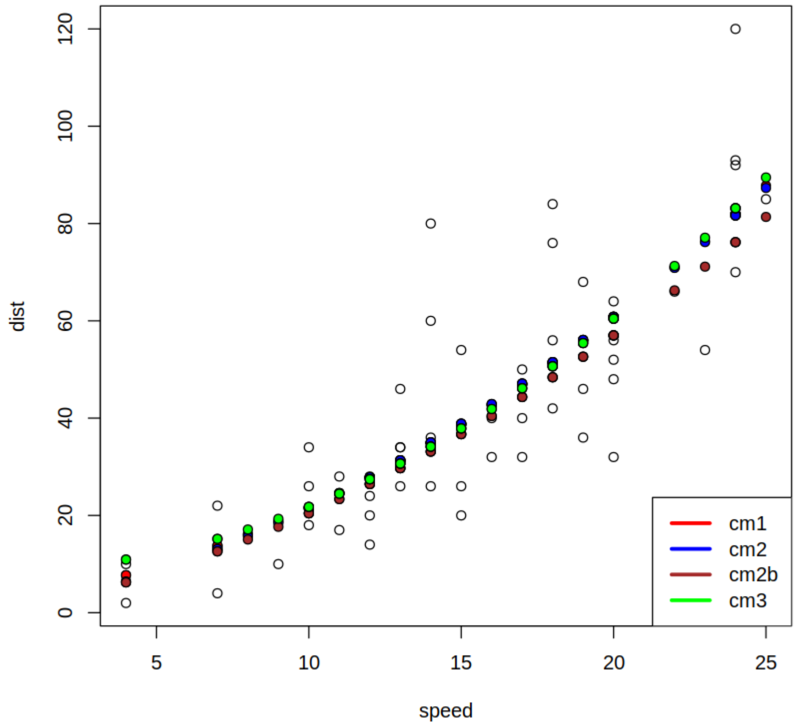
In [29]:
```
t <- (b[1]*60^2)/5280
t
```

speed: 0.842006071034171

Graficamos los resultados y guardamos

In [30]:
```
#png(filename = 'cars.png', width = 800, height = 600)
plot(cars)
points(cars$speed,cm1$fitted.values, bg = "red", pch = 21)
points(cars$speed,cm2$fitted.values, bg = "blue", pch = 21)
points(cars$speed[-c(23,35,49)],cm2b$fitted.values, bg = "brown", pch = 21)
points(cars$speed,cm3$fitted.values, bg = "green", pch = 21)
legend("bottomright", legend = c("cm1", "cm2", "cm2b","cm3"),
```

```
       lwd = 3, col = c("red", "blue","brown","green"))
#dev.off()
```



Si tenemos nuevas observaciones

> newdata = data.frame()

podemos predecir la respuesta

> predict(fit, newdata)

es decir, no necesitamos ajustar de nuevo el modelo!