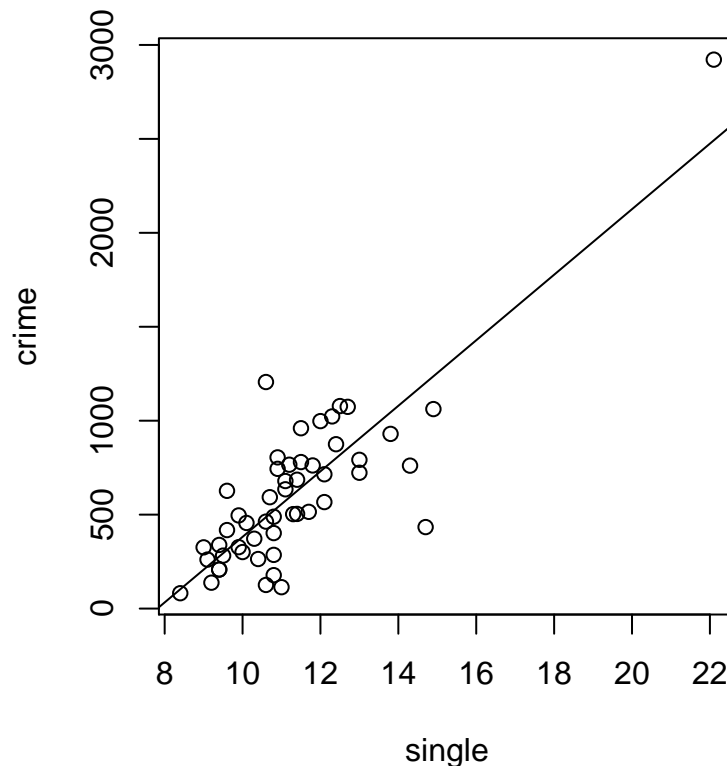


MAST20005/MAST90058: Week 6 Lab Solutions

```
library(foreign)
cdata <- read.dta("crime.dta")
```

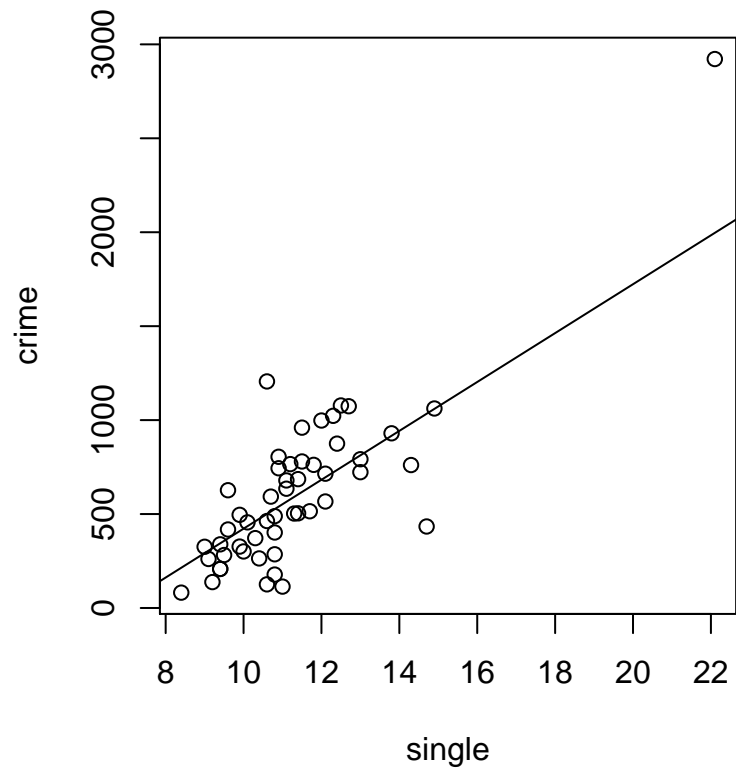
1. (a)

```
m1 <- lm(crime ~ single, cdata)
par(mar = c(4, 4, 1, 1)) # tighter figure margins
plot(crime ~ single, cdata)
abline(m1)
```

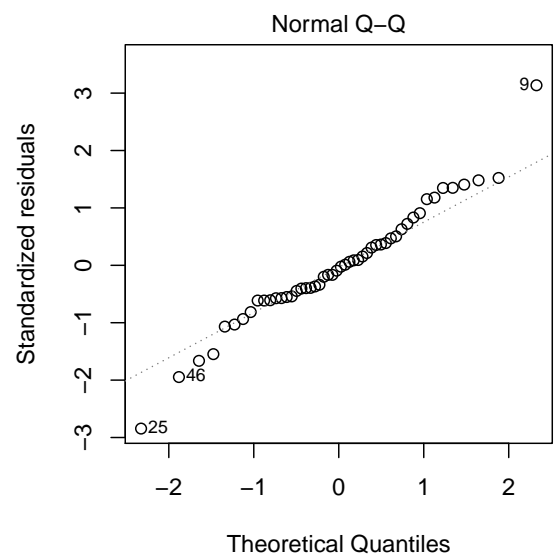
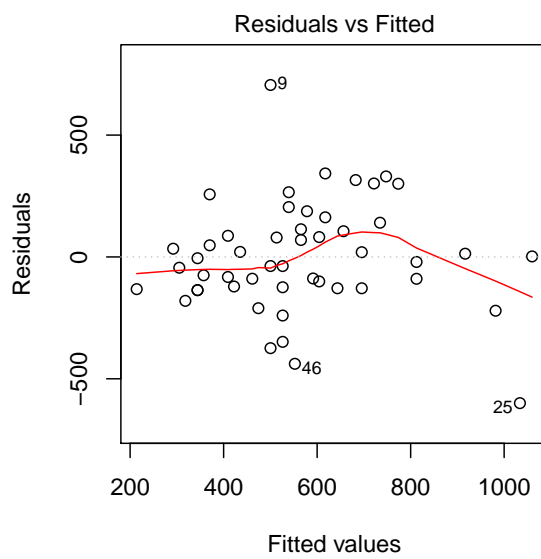


- (b) The point in the top-right corner of the plot. It corresponds to the state `dc` (Washington, D.C.).
- (c) An informal geometric argument: it seems like it might be quite influential because it is all by itself and will pull the best-fit line to be closer to itself while ‘pivoting’ it around the cluster of points in the bottom-left. (Indeed, this point does indeed have a strong influence on the parameter estimates. You will learn more about this in more advanced subjects that cover regression.)
- (d) We could do this by copying the data frame and removing the row corresponding to this outlying point. A neater way is to use the `subset` argument in `lm()`:

```
m1a <- lm(crime ~ single, cdata, subset = -51)
par(mar = c(4, 4, 1, 1)) # tighter figure margins
plot(crime ~ single, cdata)
abline(m1a)
```



```
par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
plot(m1a, 1:2)
```



```

coef(summary(m1))

##              Estimate Std. Error  t value    Pr(>|t|)
## (Intercept) -1362.5324   186.23306 -7.316276 2.150037e-09
## single      174.4186    16.16796 10.787910 1.529137e-14

coef(summary(m1a))

##              Estimate Std. Error  t value    Pr(>|t|)
## (Intercept) -878.8612   246.89537 -3.559650 8.491666e-04
## single      130.1099    22.03329  5.905152 3.498831e-07

confint(m1)

##              2.5 %    97.5 %
## (Intercept) -1736.7818 -988.2831
## single      141.9278  206.9093

confint(m1a)

##              2.5 %    97.5 %
## (Intercept) -1375.27761 -382.4448
## single      85.80902  174.4108

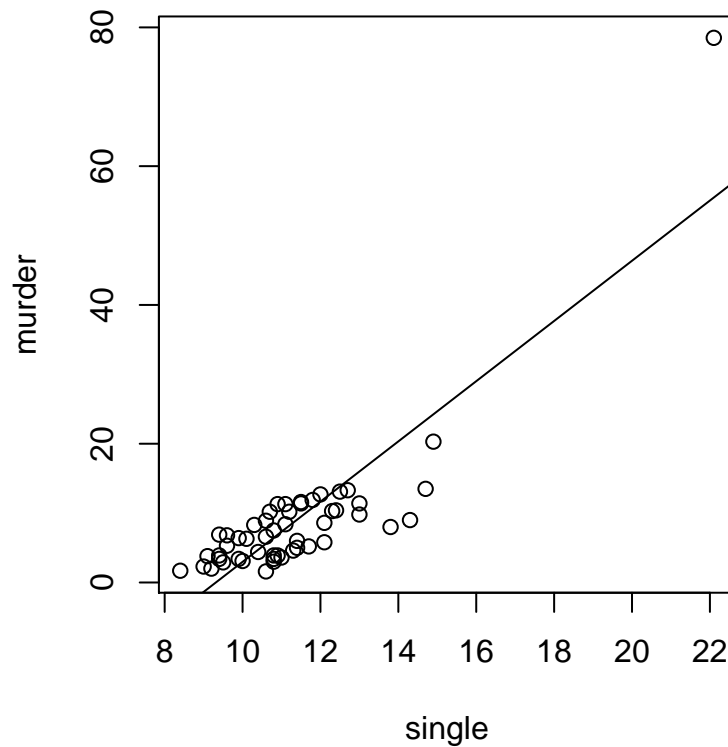
```

The estimates have shifted substantially!

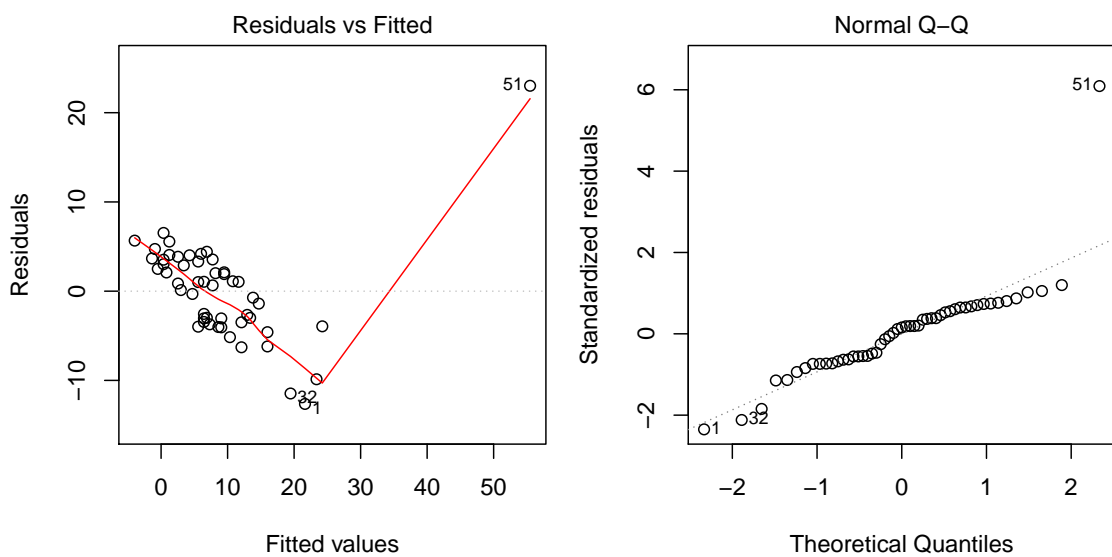
Which model we prefer to use will depend on to what extent we think that `dc` is representative or different to the other states. Given its special status within the USA and its very small size, there is a good argument for excluding it.

2. (a) `m2 <- lm(murder ~ single, cdata)`

(b) `par(mar = c(4, 4, 1, 1))` # tighter figure margins
`plot(murder ~ single, cdata)`
`abline(m2)`



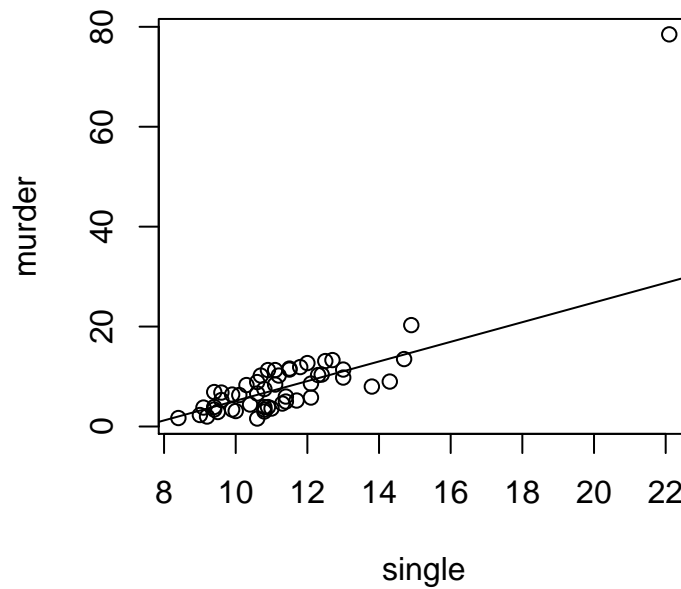
```
par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))  
plot(m2, 1:2)
```



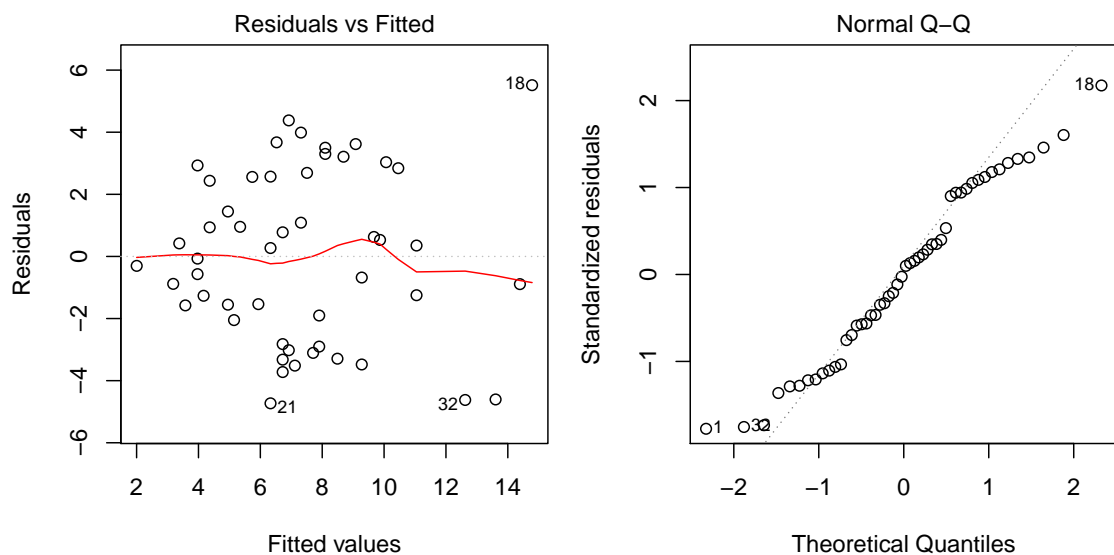
Outlying point (dc) is causing a poor model fit.

(c) Removing `dc` on the same basis as before:

```
m2a <- lm(murder ~ single, cdata, subset = -51)
plot(murder ~ single, cdata)
abline(m2a)
```



```
par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
plot(m2a, 1:2)
```



```

coef(summary(m2))

##              Estimate Std. Error   t value    Pr(>|t|)
## (Intercept) -40.41533    4.257303 -9.493177 1.099312e-12
## single      4.33913     0.369601 11.740041 7.536045e-16

coef(summary(m2a))

##              Estimate Std. Error   t value    Pr(>|t|)
## (Intercept) -14.514349    2.994406 -4.847155 1.354009e-05
## single      1.966368     0.267225  7.358474 2.078957e-09

confint(m2)

##              2.5 %    97.5 %
## (Intercept) -48.970698 -31.859958
## single      3.596389    5.081871

confint(m2a)

##              2.5 %    97.5 %
## (Intercept) -20.535005 -8.493693
## single      1.429076    2.503660

```

(d) `data12 <- data.frame(single = 12)`
`predict(m2a, newdata = data12, interval = "prediction", level = 0.9)`

```

##      fit      lwr      upr
## 1 9.082068 4.39098 13.77316

```

(e) `data8 <- data.frame(single = 8)`
`predict(m2a, newdata = data8, interval = "prediction", level = 0.9)`

```

##      fit      lwr      upr
## 1 1.216595 -3.660916 6.094107

```

Negative estimates (lower endpoint of interval)!

(f) `cdata2 <- log(cdata[, c("murder", "single")])`
`str(cdata2)`

```

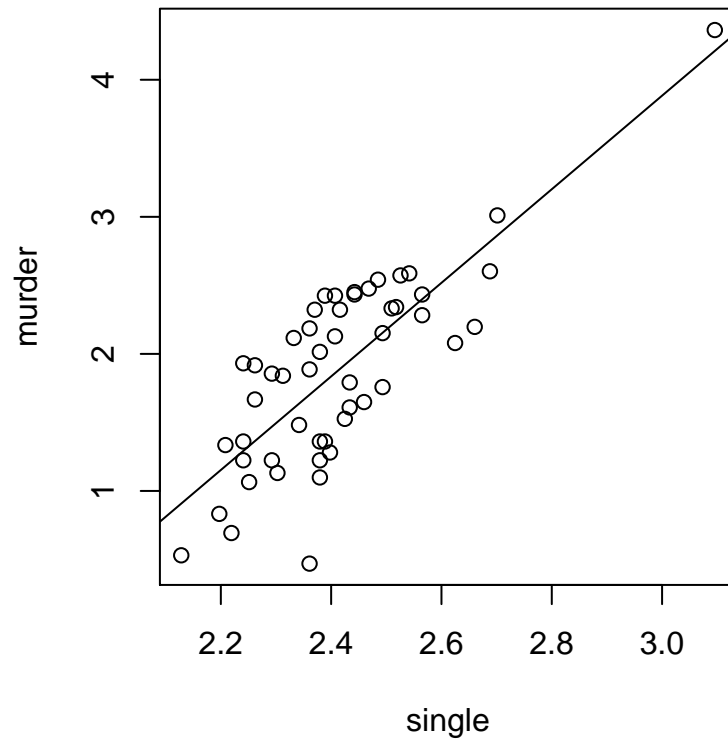
## 'data.frame': 51 obs. of  2 variables:
## $ murder: num  2.2 2.45 2.32 2.15 2.57 ...
## $ single: num  2.66 2.44 2.37 2.49 2.53 ...

```

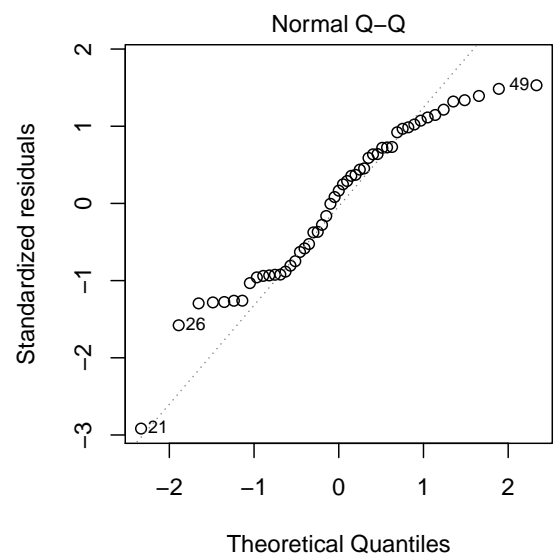
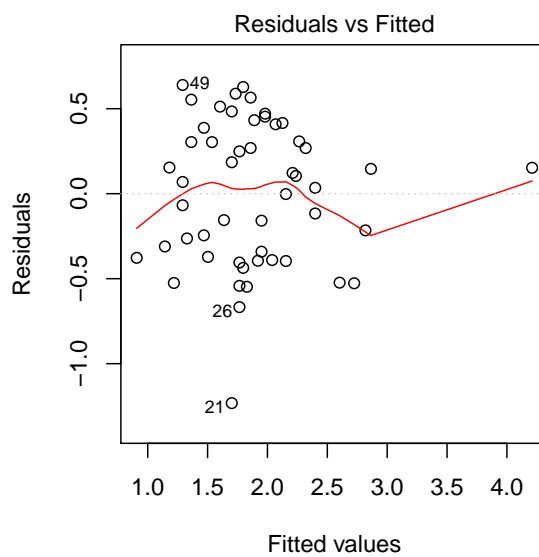
```

m3 <- lm(murder ~ single, cdata2)
par(mar = c(4, 4, 1, 1)) # tighter figure margins
plot(murder ~ single, cdata2)
abline(m3)

```



```
par(mfrow = c(1, 2), mar = c(4, 4, 2, 1))
plot(m3, 1:2)
```



```
data3 <- data.frame(single = log(8))
predint <- predict(m3, newdata = data3, interval = "prediction", level = 0.9)
predint

##          fit          lwr          upr
## 1 0.741214 -0.01148196 1.49391

exp(predint)

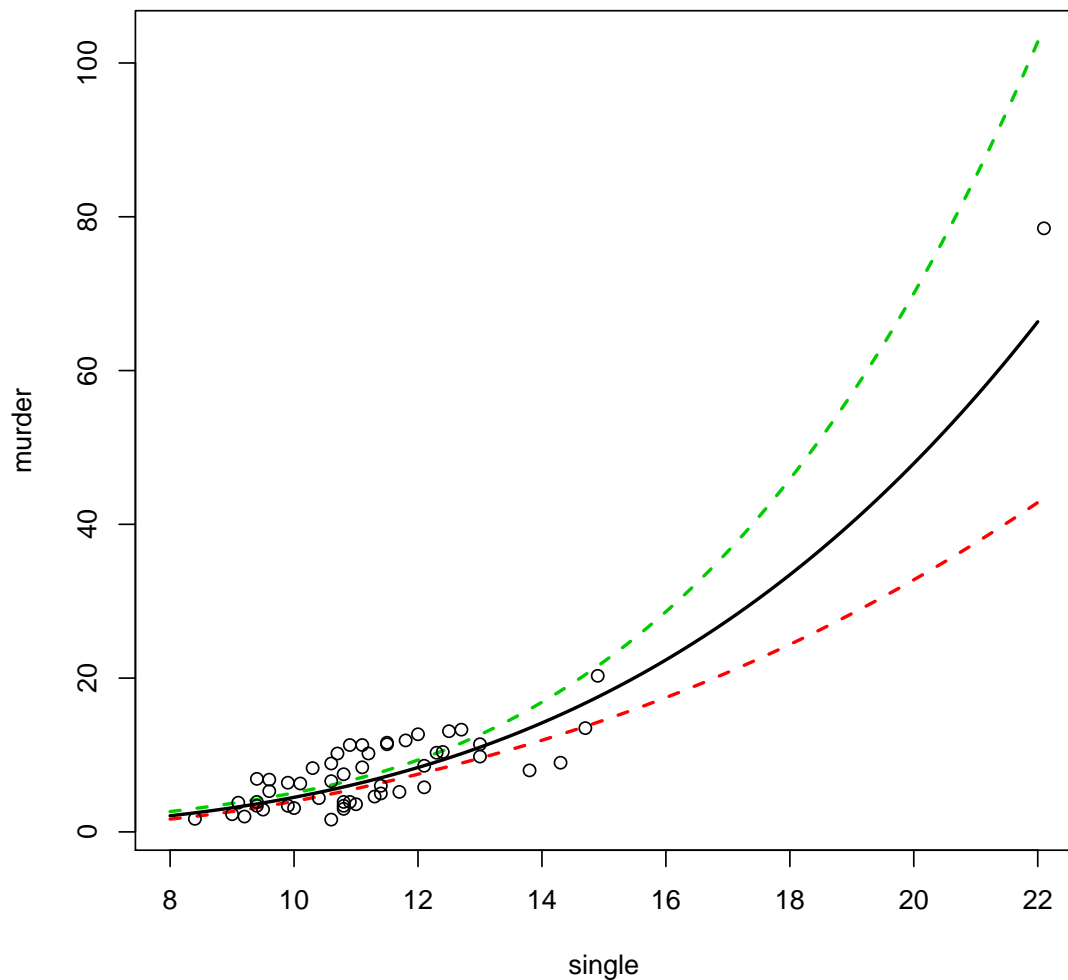
##          fit          lwr          upr
## 1 2.098482 0.9885837 4.454479
```

(g) The model equation is $\log(\text{murder}) = \alpha + \beta \log(\text{single})$, which can be expressed as:

$$\text{murder} = A \times B^{\text{single}}$$

The slope parameter β defines the multiplicative factor $B = e^\beta$, which can be interpreted as the amount by which murder rate increases multiplicatively for every percentage point increase in the proportion of single parents.

```
(h) data4 <- data.frame(single = log(seq(8, 22, 0.05)))
pred <- predict(m3, newdata = data4, interval = "confidence", level = 0.9)
matplot(exp(data4$single), exp(pred), type = "l", lty = c(1, 2, 2),
        lwd = 2, xlab = "single", ylab = "murder")
points(murder ~ single, cdata)
```



```
3. x <- c(8, 8, 8, 11, 17, 17, 20, 20, 20, 23, 26, 26, 26)
y <- c(14.8, 9.0, 11.0, 17.3, 20.8, 23.7, 24.4,
      28.9, 27.8, 29.3, 35.0, 33.4, 37.8)
m1 <- lm(y ~ x)
summary(m1)
```



```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.6821 -1.5884  0.2703  1.7767  3.1179
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.43868    1.69846   0.847   0.415
## x            1.28042    0.08982  14.255 1.95e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.161 on 11 degrees of freedom
## Multiple R-squared:  0.9486, Adjusted R-squared:  0.944
## F-statistic: 203.2 on 1 and 11 DF, p-value: 1.946e-08
```

(a) `coef(m1)`

```
## (Intercept)          x
##    1.438676    1.280423

sigma(m1)

## [1] 2.160964
```

(b) `coef(summary(m1))`

```
##              Estimate Std. Error    t value    Pr(>|t|)
## (Intercept)  1.438676  1.69846317   0.8470459 4.150186e-01
## x            1.280423  0.08982449  14.2547169 1.946385e-08
```

(c) `data2 <- data.frame(x = 18)`
`predict(m1, data2, interval = "confidence")`

```
##      fit      lwr      upr
## 1 24.48628 23.16574 25.80683
```

(d) `predict(m1, data2, interval = "prediction")`

```
##      fit      lwr      upr
## 1 24.48628 19.55012 29.42245
```