Introduction to Regression

"Regression" is a generic term for statistical methods that attempt to fit a model to data, in order to quantify the relationship between the dependent (outcome) variable and the predictor (independent) variable(s).

Assuming it fits the data reasonable well, the estimated model may then be used either to merely describe the relationship between the two groups of variables (**explanatory**), or to predict new values (**prediction**).

There are many types of regression models, here are a few most common to epidemiology:

Simple Linear Regression: Straight line regression between an outcome variable (Y) and a single explanatory or predictor variable (X).

$$E(Y) = \alpha + \beta \times X$$

Multiple Linear Regression: Same as Simple Linear Regression, but now with possibly multiple explanatory or predictor variables.

$$E(Y) = \alpha + \beta_1 \times X_1 + \beta_2 \times X_2 + \beta_3 \times X_3 + \ldots$$

A special case is polynomial regression.

$$E(Y) = \alpha + \beta_1 \times X + \beta_2 \times X^2 + \beta_3 \times X^3 + \ldots$$

Generalized Linear Model: Same as Multiple Linear Regression, but with a possibly transformed Y variable. This introduces considerable flexibility, as non-linear and non-normal situations can be easily handled.

$$G(E(Y)) = \alpha + \beta_1 \times X_1 + \beta_2 \times X_2 + \beta_3 \times X_3 + \ldots$$

In general, the transformation function $G(Y)$ can take any form, but a few forms are especially common:

- Taking $G(Y) = \text{logit}(Y)$ describes a logistic regression model:

$$\log(\frac{E(Y)}{1 - E(Y)}) = \alpha + \beta_1 \times X_1 + \beta_2 \times X_2 + \beta_3 \times X_3 + \ldots$$

- Taking $G(Y) = \log(Y)$ is also very common, leading to Poisson regression for count data, and other so called “log-linear” models.

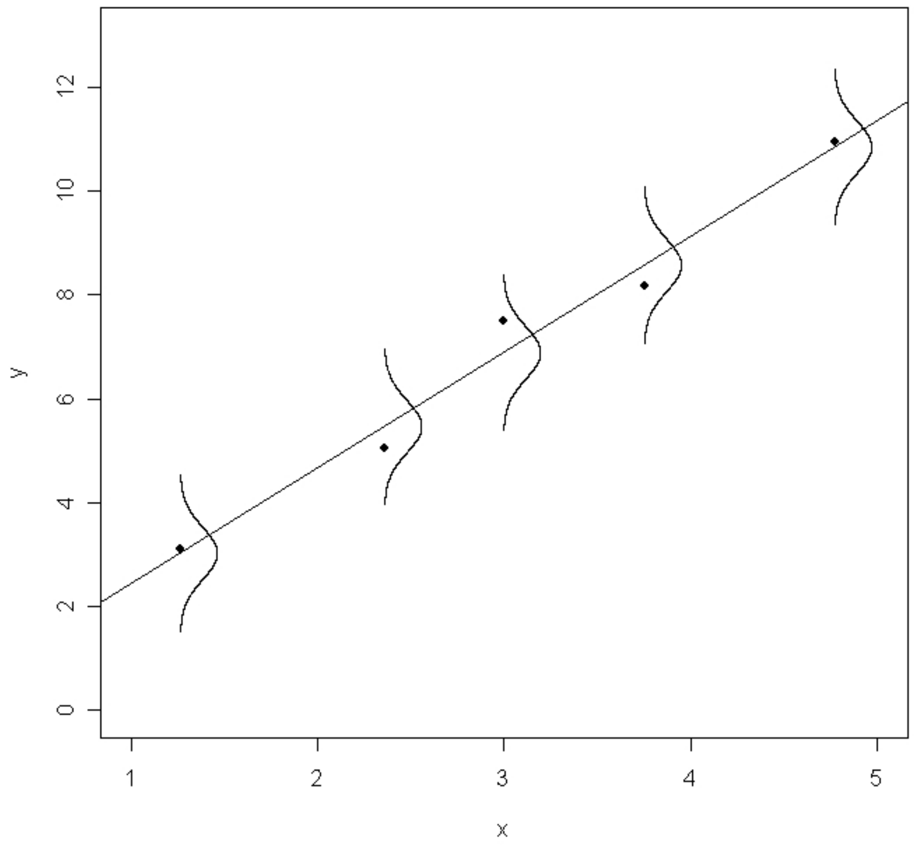
$$\log(E(Y)) = \alpha + \beta_1 \times X_1 + \beta_2 \times X_2 + \beta_3 \times X_3 + \dots$$

Multivariate Generalized Linear Model: Same as Generalized Linear Regression, but with a possibly multivariate Y variable, i.e., Y is a vector, with $Y = (Y_1, Y_2, \dots, Y_m)$. This allows several related outcomes to be modeled jointly. The β parameters become vectors in this case.

$$G(E(Y_1, Y_2, \dots, Y_m)) = \alpha + \beta_1 \times X_1 + \beta_2 \times X_2 + \beta_3 \times X_3 + \dots$$

This course focusses on linear and logistic regression.

Simple Linear Regression



Model: $Y = \alpha + \beta X +$ "error"

or equivalently: $E(Y) = \alpha + \beta X$

Assumptions:

- The "errors" (also known as "residuals") are independent $N(0, \sigma^2)$
- σ^2 is constant throughout the range
- Relationship is linear between X and Y, i.e., relation is a straight line.

Recall that the likelihood function of a model gives the likelihood that the data would be observed, given the unknown parameters, α, β, σ^2, in this case. The form of the linear relationship, together with the normality assumption for the errors, implies the following, for each data point included in the data set, $i = 1, 2 \ldots, n$:

$$Y_i \sim N(\alpha + \beta X_i, \sigma^2)$$

Recall also the form of the normal distribution:

$$f(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2\right)$$

Thus, the likelihood function contribution from a single (X_i, Y_i) pair is:

$$like(Y_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{Y_i - (\alpha + \beta X_i)}{\sigma}\right)^2\right)$$

Since each data pair (X_i, Y_i) is independent of the others (another common assumption), and since independent probabilities multiply (basic probability rule), we have the following likelihood for our model:

$$like(Y) = \prod_{i=1}^{n} \left\{ \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{Y_i - (\alpha + \beta X_i)}{\sigma}\right)^2\right) \right\}$$

Frequentist inference typically proceeds by maximizing this likelihood to find estimates of α, β and σ^2. We will omit the details, but here is a quick summary of the steps involved:

1. Take the logarithm of the likelihood function. Recall that the logarithm of a function will have the same maxima as the function itself. Because of the exponential term, the log is easier to maximize.

2. To find the values of α, β and σ^2 that maximize the logarithm of the likelihood, take the (partial) derivatives with respect to α, β and σ^2, and set these equations to zero. Solving this system of equations (3 equations in 3 unknowns) gives the maximum likelihood estimators for our unknown parameters.

If the above steps are followed, we find the following are the maximum likelihood estimates:

- $a = \hat{\alpha} = \overline{Y} - b\overline{X}$
- $b = \hat{\beta} = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^{n}(X_i - \overline{X})^2}$
- $\hat{\sigma}^2 = \sum_{i=1}^{n} \frac{(Y_i - \hat{Y}_i)^2}{n}$

where $\hat{y}_i = \hat{\alpha} + \hat{\beta} \times x_i$ is the predicted outcome for the i^{th} subject.

Statistical estimators have various desirable properties associated with them, such as unbiasedness and minimum variance. It turns out that all of the above maximum likelihood estimators are asymptotically (as sample size n goes to infinity) unbiased and minimum variance, but for finite sample sizes, while $\hat{\alpha}$ and $\hat{\beta}$ are unbiased but $\hat{\sigma}^2$ is not unbiased. Therefore, it is more common to use an unbiased (but not maximum likelihood) estimator for σ^2:

$$\hat{\sigma}^2 = \sum_{i=1}^{n} \frac{(Y_i - \hat{Y}_i)^2}{n-2}$$

Inference for Regression Parameters

Recall that confidence intervals are usually of the form:

$$\text{estimate } \pm \left\{ \begin{array}{c} z \\ t \end{array} \right\} s.d.(estimate)$$

For example, we have seen

$$\overline{x} \pm \left\{ \begin{array}{c} z \\ t \end{array} \right\} s.d.(\overline{x})$$

or

$$\overline{x} \pm \left\{ \begin{array}{c} z \\ t \end{array} \right\} \frac{s \text{ or } \sigma}{\sqrt{n}}$$

The same basic formulation is followed for inferences for regression parameters, such as α, β, or even when making predictions for future observations,

$$\hat{y}_i = \hat{\alpha} + \hat{\beta} \times x_i = a + b \times x_i$$

Since we already have point estimates for each of these items, all we are missing are the standard deviations, and what values of t or z to use.

Standard Error (Standard Deviation) Formulae

$$SE(\hat{\alpha}) = SE(a) = \sigma \sqrt{\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^{n} (x_i - \overline{x})^2}}$$

$$SE(\hat{\beta}) = SE(b) = \frac{\sigma}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2}}$$

$$SE(\text{predicted MEAN at } x) = \sigma \sqrt{\frac{1}{n} + \frac{(x - \overline{x})^2}{\sum_{i=1}^{n} (x_i - \overline{x})^2}}$$

$$SE(\text{predicted INDIVIDUAL at } x) = \sigma \sqrt{1 + \frac{1}{n} + \frac{(x - \overline{x})^2}{\sum_{i=1}^{n} (x_i - \overline{x})^2}}$$

Problem: We usually do not know σ.

Solution: Estimate σ by

$$\begin{aligned}\hat{\sigma} &= \sqrt{\frac{\text{Residual sum of squares (RSS)}}{n-2}} \\ &= \sqrt{\frac{\sum_{i=1}^{n}(y_i - \text{predicted}(y_i))^2}{n-2}} \\ &= \sqrt{\frac{\sum_{i=1}^{n}(y_i - [a + b \times x_i])^2}{n-2}}\end{aligned}$$

Confidence Intervals and Tests

Now that we know the standard errors, confidence intervals are easy to compute:

- CI for α: $\hat{\alpha} \pm t_{1-\alpha/2,n-2} \times SE(\hat{\alpha})$
- CI for β: $\hat{\beta} \pm t_{1-\alpha/2,n-2} \times SE(\hat{\beta})$
- CI for predicted mean:
  $\hat{y_i} \pm t_{1-\alpha/2,n-2} \times SE(\text{predicted MEAN at } x)$
- CI for predicted individual:
  $\hat{y_i} \pm t_{1-\alpha/2,n-2} \times SE(\text{predicted INDIVIDUAL at } x)$

Even though we will rarely use hypothesis tests after we know the CIs, for completeness, tests of hypotheses about α and β can be constructed as follows:

To test $H_0 : \ \alpha = \alpha_0$, use the fact that

$$\frac{\hat{\alpha} - \alpha_0}{SE(\hat{\alpha})} \sim t_{n-2},$$

and similarly for β:

To test $H_0 : \ \beta = \beta_0$, use the fact that

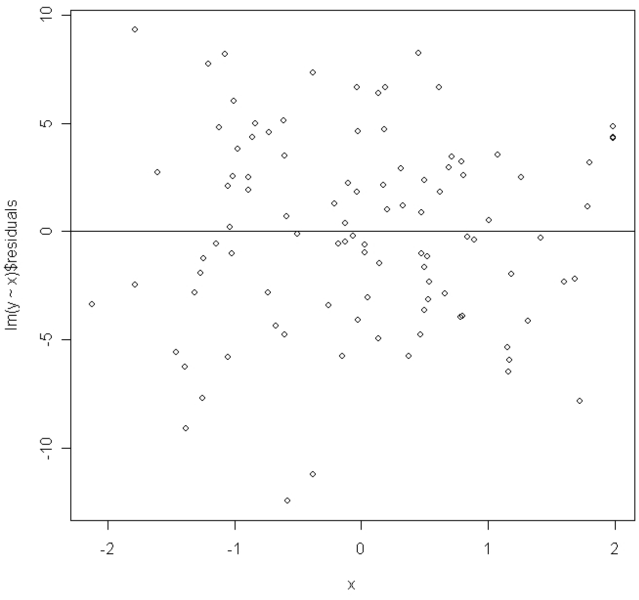
$$\frac{\hat{\beta} - \beta_0}{SE(\hat{\beta})} \sim t_{n-2}.$$

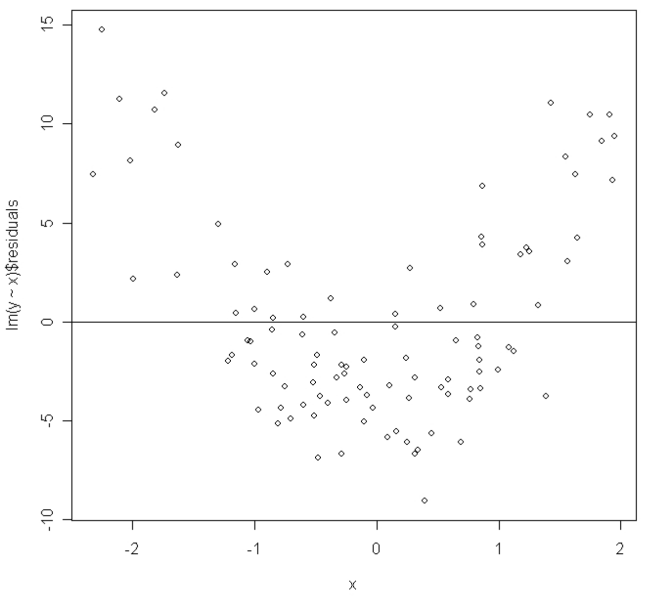
Residuals and Assumptions of Simple Linear Regression

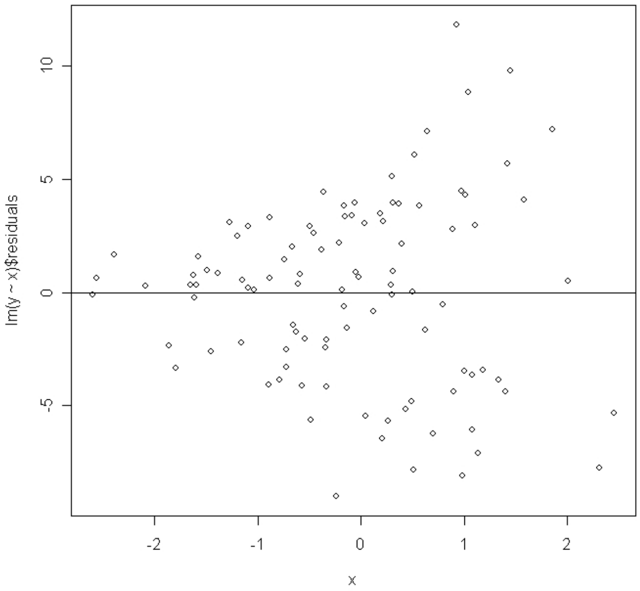
Recall the three main assumptions of simple linear regression:

- "error" is $N(0, \sigma^2)$
- σ^2 is constant throughout the range
- Relationship is linear between X and Y, i.e., relation is a straight line.

We can roughly verify if these assumptions hold true by looking at patterns of residuals. Let's look at three examples:







In the first case, all assumptions seem satisfied, but in the second graph, the relationship does not seem linear, and in the third, the variance is not constant throughout the range.

Example of Simple Linear Regression

We continue with an example: The data below describe the tooth decay experience of 7257 children 12–14 years old in 21 communities according to the fluoride concentration of their public water supply. DMF denotes "Decayed, Missing or Filled."

Community Number	DMF per 100 children	Fluoride Concentration in ppm
1	236	1.9
2	246	2.6
3	252	1.8
4	258	1.2
5	281	1.2
6	303	1.2
7	323	1.3
8	343	0.9
9	412	0.6
10	444	0.5
11	556	0.4
12	652	0.3
13	673	0.0
14	703	0.2
15	706	0.1
16	722	0.0
17	733	0.2
18	772	0.1
19	810	0.0
20	823	0.1
21	1027	0.1

A typical simple linear regression analysis will follow these steps:

(a) Draw a rough scatter plot to visually examine the association between DMF teeth and fluoride.
(b) Calculate the parameters of regression line of DMF teeth on fluoride concentration.
(c) Estimate of σ, the residual standard deviation, and thus calculate 95%

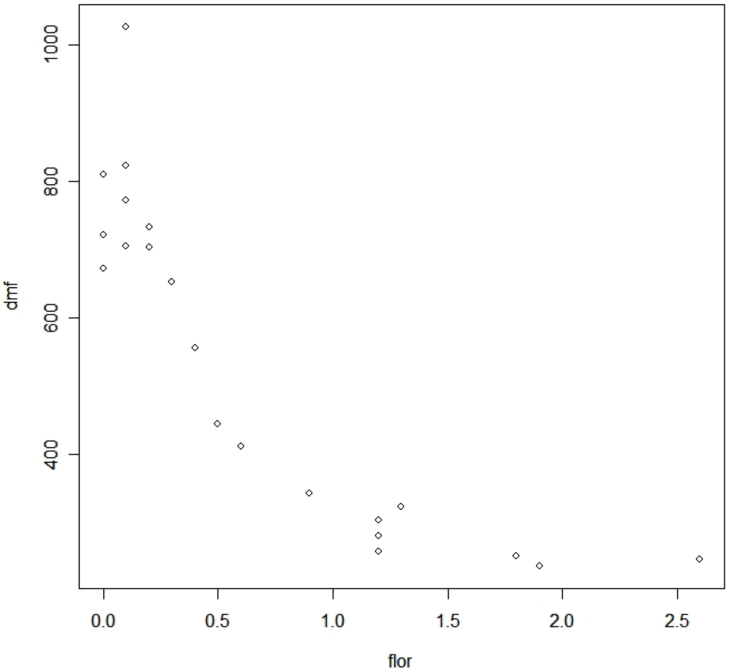
confidence intervals for the intercept and slope parameters. With these estimates available, clinically interpret the results.
(d) Make predictions, for example for the average number of DMF teeth there would be in a community with a fluoride concentration of 1.5 ppm.
(e) Calculate a 95% confidence interval around your answer in (d).
(f) Examine the graph of the residuals. Does a linear regression seem appropriate?

We will now go through each of these steps:

(a) Using the following R program with results:

```
> dmf.data <- data.frame(
dmf = c( 236, 246, 252, 258, 281, 303, 323, 343, 412, 444, 556, 652,
673, 703, 706, 722, 733, 772, 810, 823, 1027),
flor =  c( 1.9, 2.6, 1.8, 1.2, 1.2, 1.2, 1.3, 0.9, 0.6, 0.5, 0.4, 0.3,
0.0, 0.2, 0.1, 0.0, 0.2, 0.1, 0.0, 0.1, 0.1))
> dmf.data     #  just to look at the data set
    dmf  flor
1   236 1.9
2   246 2.6
3   252 1.8
4   258 1.2
5   281 1.2
6   303 1.2
7   323 1.3
8   343 0.9
9   412 0.6
10  444 0.5
11  556 0.4
12  652 0.3
13  673 0.0
14  703 0.2
15  706 0.1
16  722 0.0
17  733 0.2
18  772 0.1
19  810 0.0
20  823 0.1
21 1027 0.1
```

```
> attach(dmf.data)  #  For convenience, to make the dmf and flor variables
>                   #  available outside the data.frame
>
> plot(flor, dmf)   #  Create a scatter plot
```



(b) Again we can use R, obtaining:

```
> regression.out<-lm(dmf~flor)
> summary(regression.out)

Call:
lm(formula = dmf ~ flor)

Residuals:
     Min       1Q   Median       3Q      Max
-152.825  -94.305    1.575   56.495  322.575

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   732.34      38.55  18.996  8.1e-14 ***
flor         -279.20      38.18  -7.312  6.2e-07 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 127.3 on 19 degrees of freedom
Multiple R-Squared: 0.7378,     Adjusted R-squared: 0.724
F-statistic: 53.47 on 1 and 19 DF,  p-value: 6.199e-07
```

(c) From the printout above, the residual standard error is $\sigma = 127.3$.

Note that the R summary does not provide confidence intervals, although there is a separate function called `confint` that will produce confidence intervals following just about any regression analysis.

```
> confint(regression.out)
                2.5 %    97.5 %
(Intercept)  651.6540  813.0350
flor        -359.1138 -199.2855
```

It is instructive to see how these are done by hand as well: We have seen the formulae for CIs above, and R provides all of the info needed by these formulae, so that confidence intervals can be obtained “manually”. This can be done on a hand calculator, but of course it is easier to do this in R itself:

The printout above gives us the three numbers we need to construct the confidence interval, namely the coefficient estimates, the standard errors for each parameter, and the degrees of freedom. The same three numbers are also needed for hypothesis tests about regression coefficients, although these will be used only rarely in this course (and, of course, the summary output above already gives you the p-values!).

Note that if we want a 95% confidence interval, the critical value for 19 degrees for freedom is given by

```
> qt(0.975, df= 19)
[1] 2.093024
```

So, we calculate:

```
> 732.34 + c(-1, 1) * qt(0.975, df= 19) * 38.55
[1] 651.6539 813.0261
```

as the 95% CI for the intercept α, and

```
> -279.20 + c(-1, 1) * qt(0.975, df= 19)* 38.18
[1] -359.1117 -199.2883
```

as the CI for the slope β.

Clinical interpretation: At a 0 concentration level for fluoride, “we are confident that” the true DMF level is between 651.7 and 813.0. For each rise of one unit in fluoride, “we are confident that” the DMF level *decreases* by an amount somewhere between -359.1 and -199.3.

Because we will be calculating confidence intervals so often in this course, it is worthwhile to spend a few minutes to automate the above process, all in one function. First, notice this about how regression in R works:

```
> summary(regression.out)$coefficients
             Estimate Std. Error   t value     Pr(>|t|)
(Intercept)  732.3445   38.55211 18.996224 8.103719e-14
flor        -279.1996   38.18119 -7.312492 6.198645e-07

> summary(regression.out)$coefficients[1:8]
[1]  7.323445e+02 -2.791996e+02  3.855211e+01  3.818119e+01
[5]  1.899622e+01 -7.312492e+00  8.103719e-14  6.198645e-07
```

So, of interest to us are coefficients [1], [2], [3], and [4], corresponding to the coefficient values for α and β, and their standard errors. Thus, we can create the following new function in R:

```
regression.with.ci <- function(regress.out, level=0.95)
{
```

```
################################################################
#                                                              #
#  This function takes the output from an lm                   #
#  (linear model) command in R and provides not                #
#  only the usual output from the summary command, but         #
#  adds confidence intervals for intercept and slope.          #
#                                                              #
################################################################
usual.output <- summary(regress.out)
t.quantile <- qt(1-(1-level)/2, df=regress.out$df)
intercept.ci <- summary(regress.out)$coefficients[1]
        + c(-1, 1) * t.quantile * summary(regress.out)$coefficients[3]
slope.ci <- summary(regress.out)$coefficients[2]
       + c(-1, 1) * t.quantile * summary(regress.out)$coefficients[4]
output <- list(regression.table = usual.output, intercept.ci = intercept.ci,
                slope.ci = slope.ci)
return(output)
}
```

Note that this returns exactly the same results as we calculated previously, either “manually” or using confint:

```
> regression.with.ci(regression.out)
$regression.table

Call:
lm(formula = dmf ~ flor)

Residuals:
     Min       1Q   Median       3Q      Max
-152.825  -94.305    1.575   56.495  322.575

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   732.34      38.55  18.996  8.1e-14 ***
flor         -279.20      38.18  -7.312  6.2e-07 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
Residual standard error: 127.3 on 19 degrees of freedom
Multiple R-Squared: 0.7378,     Adjusted R-squared: 0.724
F-statistic: 53.47 on 1 and 19 DF,  p-value: 6.199e-07

$intercept.ci
[1] 651.654 813.035

$slope.ci
[1] -359.1138 -199.2855
```

Suggestion: Cut and paste this function into your version of R, to have this available for future analyses (or wait for the more general multivariate version discussed in the next lecture).

(d) Again, we have enough information in the R outputs to plug into the prediction formulae given above, but it is very easy to make predictions in R. First, fitted values are immediately available for all X values (fluoride, in this case), once a regression is run:

```
> flor
[1] 1.9 2.6 1.8 1.2 1.2 1.2 1.3 0.9 0.6 0.5 0.4 0.3 0.0
    0.2 0.1 0.0 0.2 0.1 0.0 0.1 0.1
> regression.out$fitted.values
         1          2          3          4          5          6          7
201.865194   6.425445 229.785158 397.304942 397.304942 397.304942 369.384978
         8          9         10         11         12         13         14
481.064834 564.824726 592.744690 620.664654 648.584618 732.344510 676.504582
        15         16          17         18         19         20         21
704.424546 732.344510  676.504582 704.424546 732.344510 704.424546 704.424546
```

However, we notice that the value 1.5 is not on the original list, so we use:

```
> predict.lm(regression.out, newdata=data.frame(flor=1.5))
[1] 313.5450
```

(e) We can also get confidence intervals for these predictions:

```
> predict.lm(regression.out, newdata=data.frame(flor=1.5), interval="prediction")
          fit      lwr      upr
[1,] 313.5450 33.39467 593.6954
> predict.lm(regression.out, newdata=data.frame(flor=1.5), interval="confidence")
          fit      lwr      upr
[1,] 313.5450 227.1223 399.9678
```

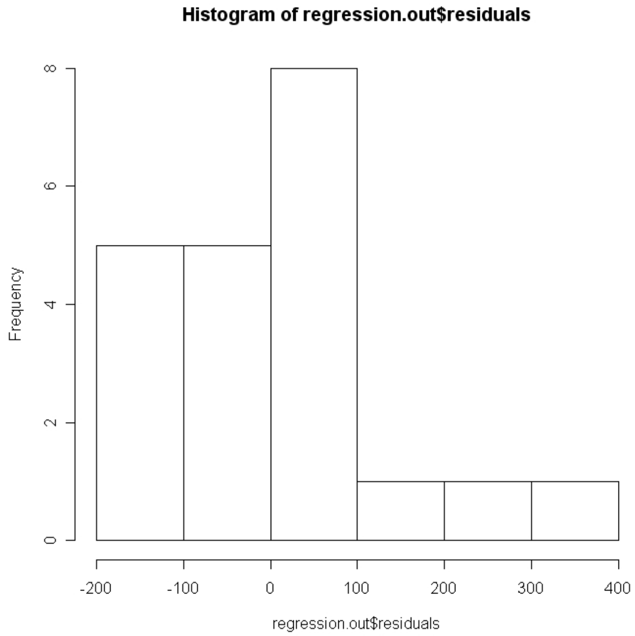
The first interval is a prediction for the "next town" with a fluoride concentration of 1.5, while the second interval is the confidence interval for the mean prediction (i.e., for an infinite number of towns, each with value 1.5 …note that second interval is much smaller than first).

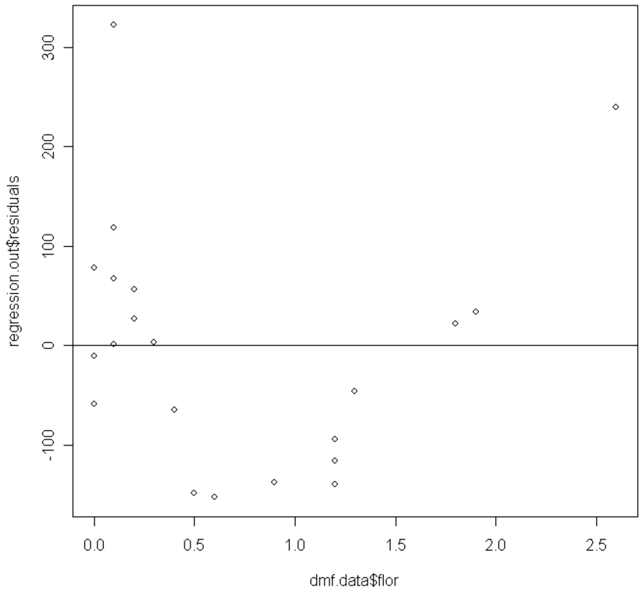
(e) The residuals are always immediately available:

```
> regression.out$residuals
          1           2           3           4           5           6           7
  34.134806  239.574555   22.214842 -139.304942 -116.304942  -94.304942  -46.384978
          8           9          10          11          12          13          14
-138.064834 -152.824726 -148.744690  -64.664654    3.415382  -59.344510   26.495418
         15          16          17          18          19          20          21
   1.575454  -10.344510   56.495418   67.575454   77.655490  118.575454  322.575454
```

so we can graph them:

```
> hist(regression.out$residuals)
> plot(dmf.data$flor, regression.out$residuals)
> abline(h=0)
```





Note that the fit is actually quite poor, the function is not linear. A quadratic model (i.e., with a flor squared term) may fit better, and we will return to this example later.