Simple Linear Regression with R

Getting and Opening Data Files

We will use an example data set from *Regression Analysis by Example* (4th ed.) by Chatterjee and Hadi (Wiley, New York, 2006). Go to the web site for this book at http://www.ilr.cornell.edu/~hadi/rabe4/. We will use the computer repair data. In this study a random sample of service call records for a computer repair operation were examined and the length of each call (in minutes) and the number of components repaired or replaced were recorded. The data are in file P027.txt. Follow the directions on the book's home page to download this and save it in the R folder on your computer. Then read the file into R as shown below. (`header=TRUE` means the file contains names for the variables on the first line.)

```
> repairs = read.table("P027.txt",header=TRUE)
> attach(repairs)
> repairs
   Minutes Units
1       23     1
2       29     2
3       49     3
4       64     4
5       74     4
6       87     5
7       96     6
8       97     6
9      109     7
10     119     8
11     149     9
12     145     9
13     154    10
14     166    10
```

Simple Plots for Each Variable

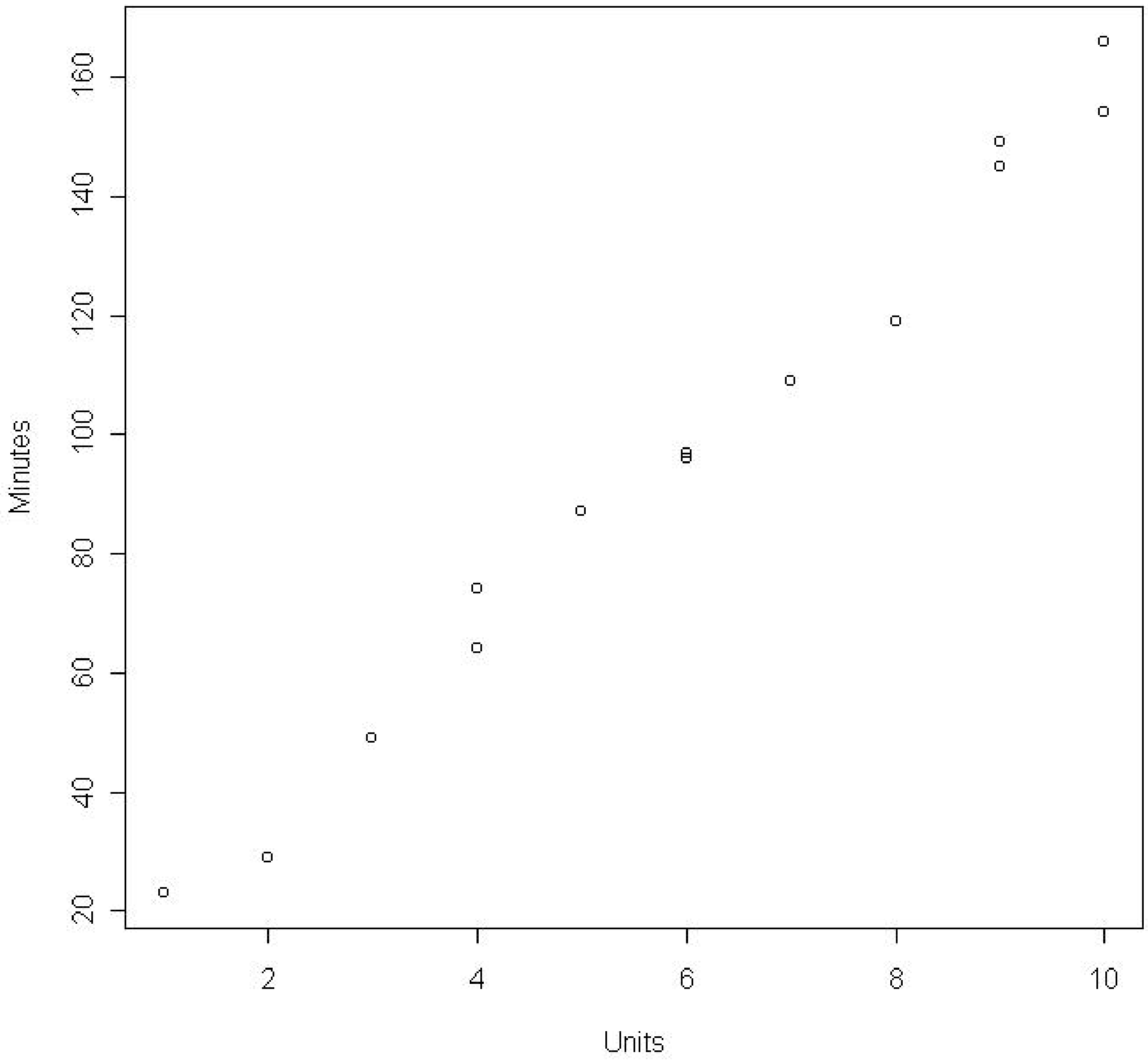
Of course, the first step is to look at your data.

```
> stem(Minutes)

  The decimal point is 2 digit(s) to the right of the |

  0 | 23
  0 | 5679
  1 | 0012
  1 | 5557

> stem(Units)

  The decimal point is at the |

   0 | 0
   2 | 00
   4 | 000
   6 | 000
   8 | 000
  10 | 00
```

We could have made histograms or boxplots. We simply want to see if there are any peculiarities in the data for each variable by itself before we look into relationships between variables. We see none here.

Scatterplots

```
> plot(Units,Minutes)
```



Note that the first variable in the `plot` command is plotted on the horizontal axis. We are not surprised to see that the length of a service call increases with the number of components repaired or replaced.

Correlation and Covariance

```
> cor(Units,Minutes)
[1] 0.9936987
> cov(Units,Minutes)
[1] 136
```

Running the Regression

The regression command is `lm` for linear model. We will store that model in a variable called `model`. The order of the variables is dependent followed by a tilde "~" followed by a list of independent variables.

```
> model = lm(Minutes ~ Units)
> model

Call:
lm(formula = Minutes ~ Units)

Coefficients:
(Intercept)        Units
      4.162       15.509

> summary(model)

Call:
lm(formula = Minutes ~ Units)

Residuals:
    Min      1Q  Median      3Q     Max
-9.2318 -3.3415 -0.7143  4.7769  7.8033

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.162      3.355    1.24    0.239
Units         15.509      0.505   30.71 8.92e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.392 on 12 degrees of freedom
Multiple R-Squared: 0.9874,     Adjusted R-squared: 0.9864
F-statistic: 943.2 on 1 and 12 DF,  p-value: 8.916e-13
```

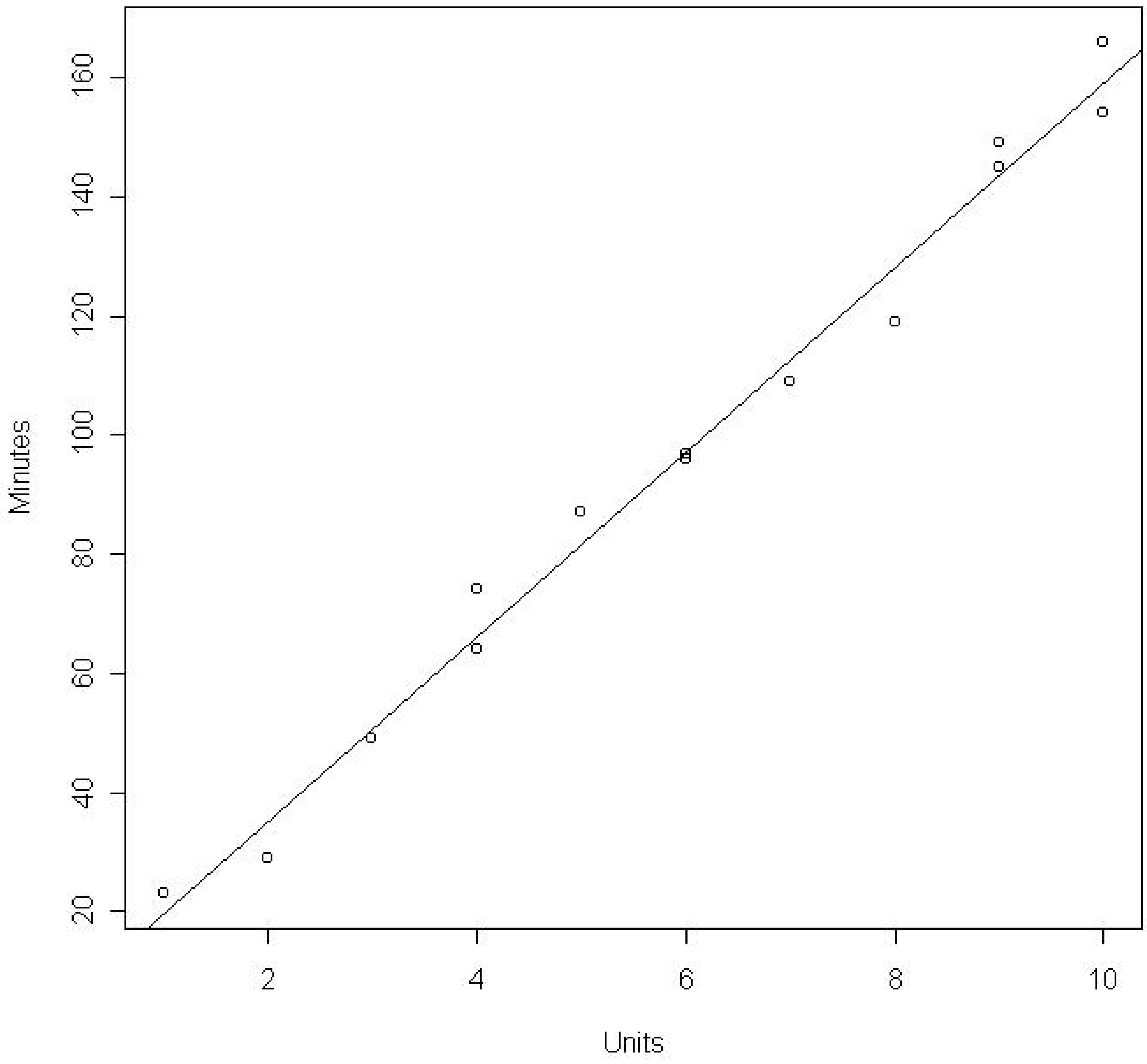
The regression equation is *minutes* = 4.162 + 15.509**units*. The "*t* values" test the hypotheses that the corresponding population parameters are 0. Usually we test whether the slope is zero because if it is then the model is not much use. Here the

p

-value for that test is "8.92e-13" which is to say 8.92X10^{-13} or 0.000000000000892, so we would reject the hypothesis that the slope is zero. If you wish to test a nonzero value, subtract it from the coefficient in the regression output (15.509) and divide the result by the coefficient's s.e. (0.505). (Use a calculator for this.) Similarly, if you want confidence intervals, use the coefficient plus or minus the product of its s.e. with a *t*-value for the desired confidence level and 12 degrees of freedom. (Use a calculator for this.) This also works for the intercept (4.162) using its s.e. (3.355).

To plot the regression line on the scatterplot, type

```
> abline(model)
```



You can cut and paste R output into your own reports but note that the text windows on the statistics.com Assignments page will only accept text input. So, of the output examples above, the scatterplots could *not* be pasted there. All the text that appears showing our interaction with R *can* be pasted into Assignments. To copy the contents of a graphics window (say for a report you are writing with your word processor), first click on File in the graph window, then select any of the first three options.

Regression through the Origin

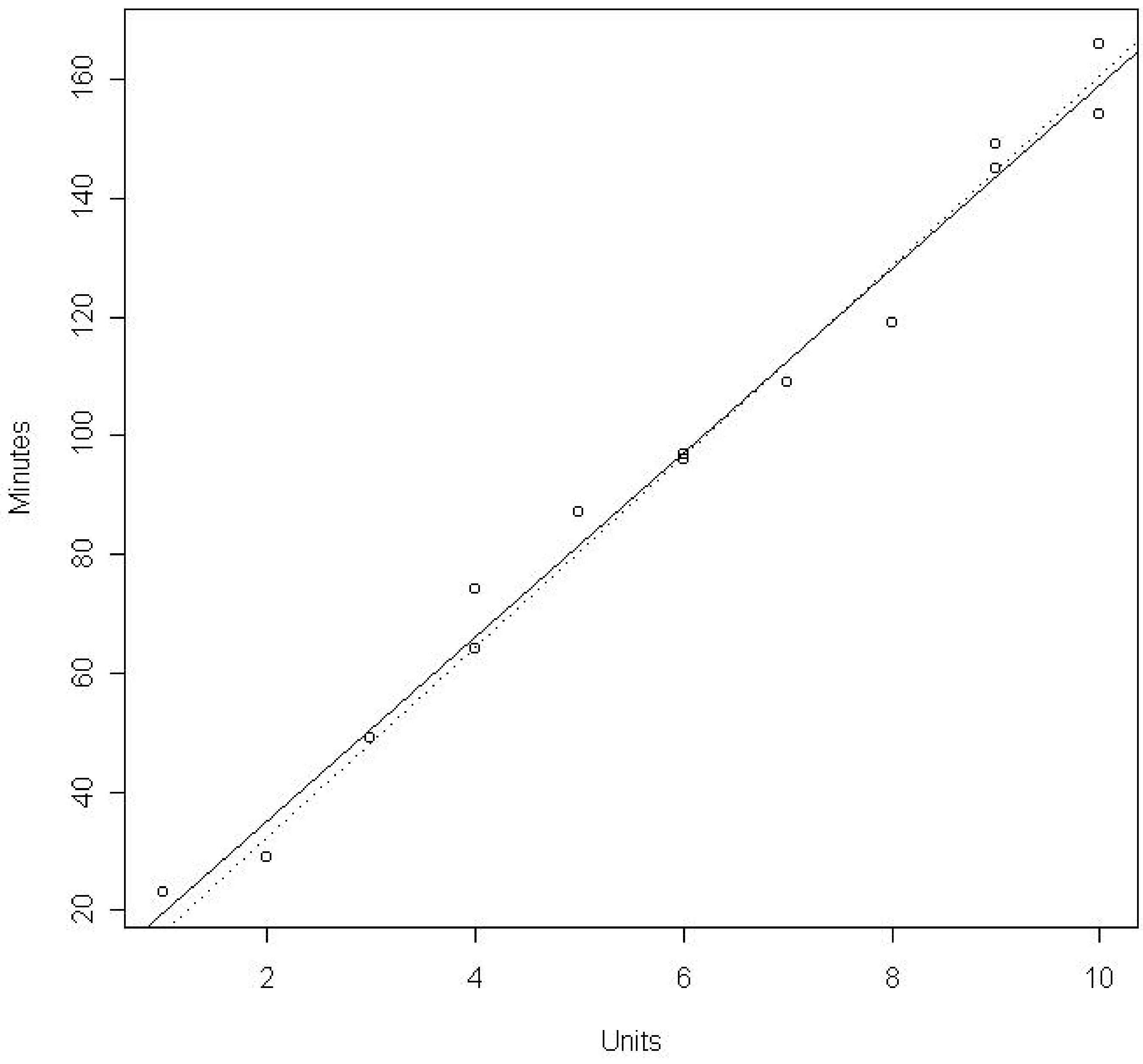
To fit a regression line through the origin (i.e., intercept=0) redo the regression but this time include that 0 in the model specification.

```
> model2 = lm(Minutes ~ 0 + Units)
> summary(model2)

Call:
lm(formula = Minutes ~ 0 + Units)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-9.5955 -2.4733  0.4417  5.0243  9.7023 

Coefficients:
      Estimate Std. Error t value Pr(>|t|)    
Units  16.0744     0.2213   72.63   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 5.502 on 13 degrees of freedom
Multiple R-Squared: 0.9975,     Adjusted R-squared: 0.9974 
F-statistic:  5274 on 1 and 13 DF,  p-value: < 2.2e-16
```

or *minutes* = 16.0744**units*. We can add this line to the graph to see how different it is.

```
> abline(model2, lty = "dotted")
```



Not much.

Predictions

We predicted the length of a service call with four components repaired or replaced, then got confidence intervals for the prediction of a single observation and for the mean of all observations with `Units`=4 (and based on our first model).

```
> predict(model, newdata=data.frame(Units=4))
[1] 66.19674
> predict(model, newdata=data.frame(Units=4), interval = "pred")
          fit      lwr      upr
[1,] 66.19674 53.83936 78.55413
> predict(model, newdata=data.frame(Units=4), interval = "confidence")
          fit      lwr      upr
[1,] 66.19674 62.36271 70.03077
```

The syntax is tortured and will not be explained here.