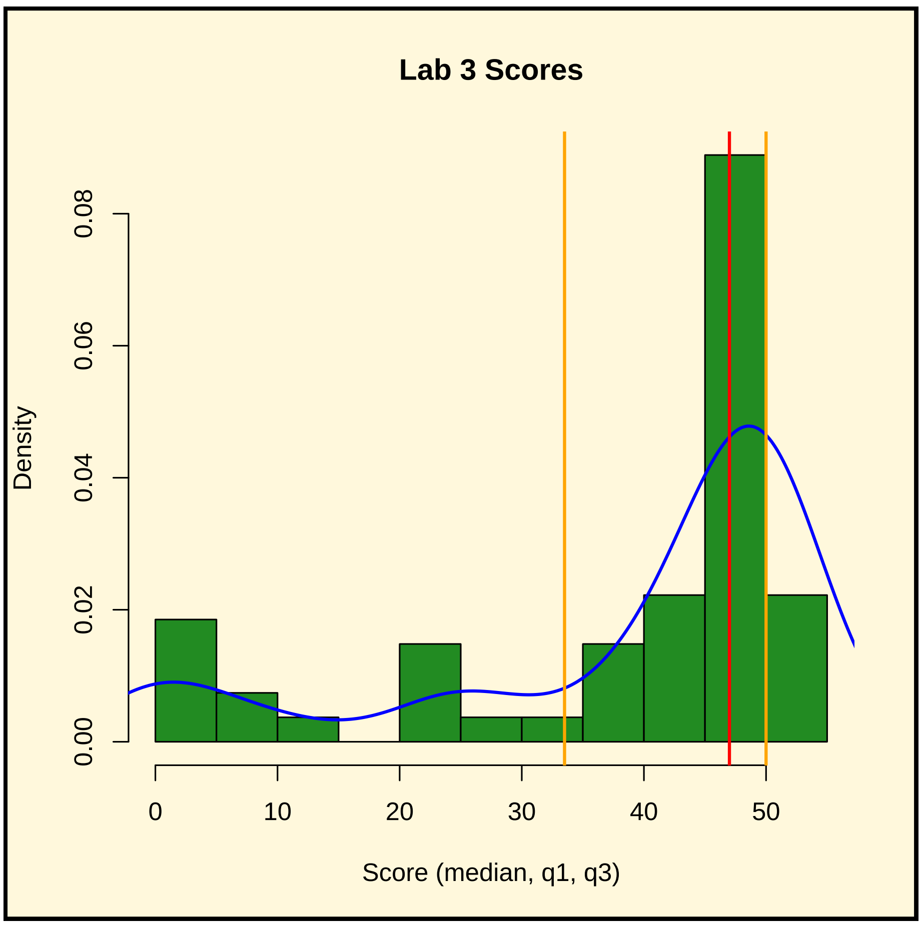
ISTA 116 Lab: Week 9

Last Revised October 17, 2011

1 HW 3



2 Midterm

- **Review the Midterm**

3 Simple Linear Regression Model

Used to describe paired data sets that are related in a linear manner.

A linear relationship between variables x and y means that $y = mx + b$, where m is the slope of the line and b the intercept. We call x the *independent* variable and y the *dependent* one.

We don't assume these variables have an *exact* linear relationship; the possibility for noise or error is taken into account.

To fit the linear regression ("least-squares") model to data, we pass the linear model desired[1] to the `lm()` function:

```
> lm(y ~ x)
```

The formula `y ~ x` ("y as a function of x") implies the linear relationship between y (dependent/response variable) and x (independent/predictor variable).

The `lm()` function creates a linear model object from which a wealth of information can be extracted.

3.1 Example 1

Consider the `cars` dataset. The data give the speed (`speed`) of cars and the distances (`dist`) taken to come to a complete stop. Here, we will fit a linear regression model using `speed` as the independent variable and `dist` as the dependent variable (these variables should be plotted first to check for evidence of a linear relation).

```
> data(cars)
> attach(cars)
> plot(speed, dist)
```

To compute the least-squares:

```
> fit <- lm(dist ~ speed)
```

[1] other models are possible, we only look at simple linear models here

The object `fit` is a linear model object. To see what it contains, type:

```
> attributes(fit)
```

To get the least squares estimates of the slope and intercept, type:

```
> fit
```

The fitted regression model has an intercept of -17.579 and a slope of 3.932.

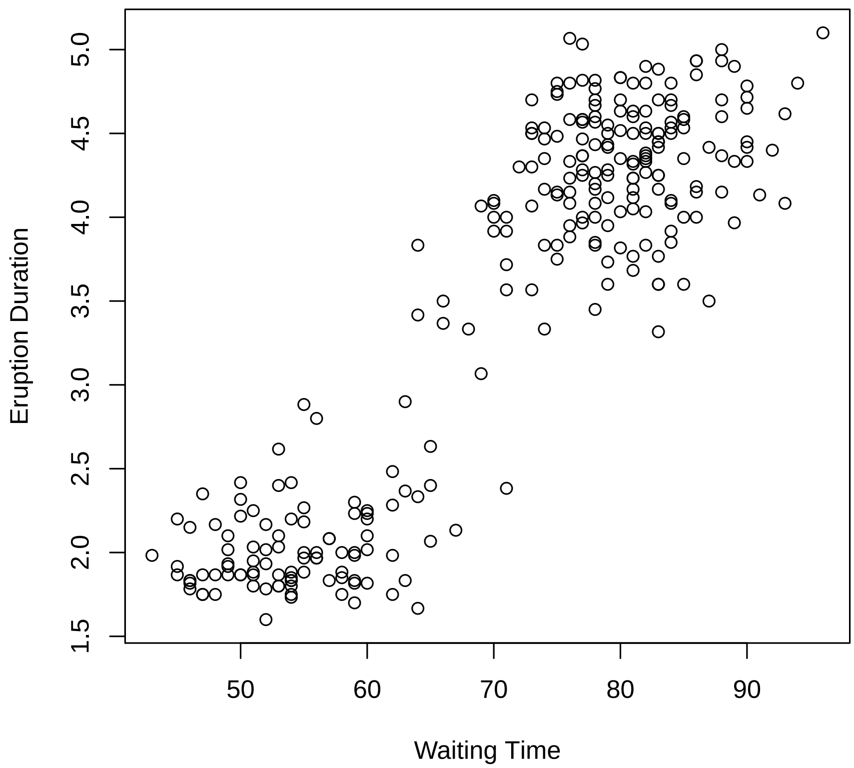
We can add the fitted regression line to a scatterplot:

```
> plot(speed, dist)
> abline(fit)
```

3.2 Example 2

The data set *faithful* contains sample data of two random variables named *waiting* and *eruptions*.

```
> plot(faithful$waiting, faithful$eruptions, xlab = "Waiting Time",
+     ylab = "Eruption Duration")
```



The *waiting* variable denotes the waiting time until the next eruption, and *eruptions* denotes the duration. Its linear regression model can be expressed as:

$$\hat{Eruptions_i} = \hat{\beta}_0 + \hat{\beta}_1 * Waiting_i$$

3.2.1 Making a Prediction

Apply a simple linear regression model for the data set *faithful*, and estimate the next eruption duration if the waiting time since the last eruption has been 80 minutes.

We apply the `lm` function to a formula that describes the variable eruptions by the variable waiting, and save the linear regression model in a new variable eruption.lm.

```
> eruption.lm = lm(eruptions ~ waiting, data = faithful)
```

Then we extract the parameters of the estimated regression equation with the coefficients function.

```
> (coeffs = coefficients(eruption.lm))
```

```
(Intercept)     waiting
-1.87401599  0.07562795
```

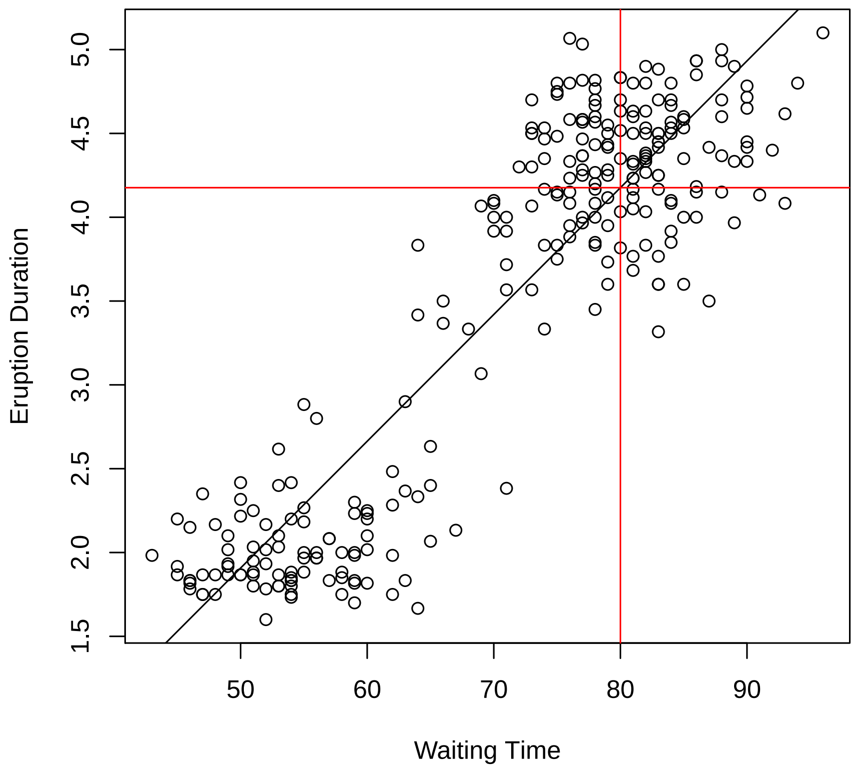
We now fit the eruption duration using the estimated regression equation.

```
> wait_time = 80
> (duration = coeffs[1] + coeffs[2] * wait_time)
```

```
(Intercept)
    4.17622
```

Based on the simple linear regression model, if the waiting time since the last eruption has been 80 minutes, we expect the next one to last 4.1762 minutes:

```
> plot(faithful$waiting, faithful$eruptions, xlab = "Waiting Time",
+     ylab = "Eruption Duration")
> abline(eruption.lm)
> abline(v = wait_time, h = duration, col = "red")
```



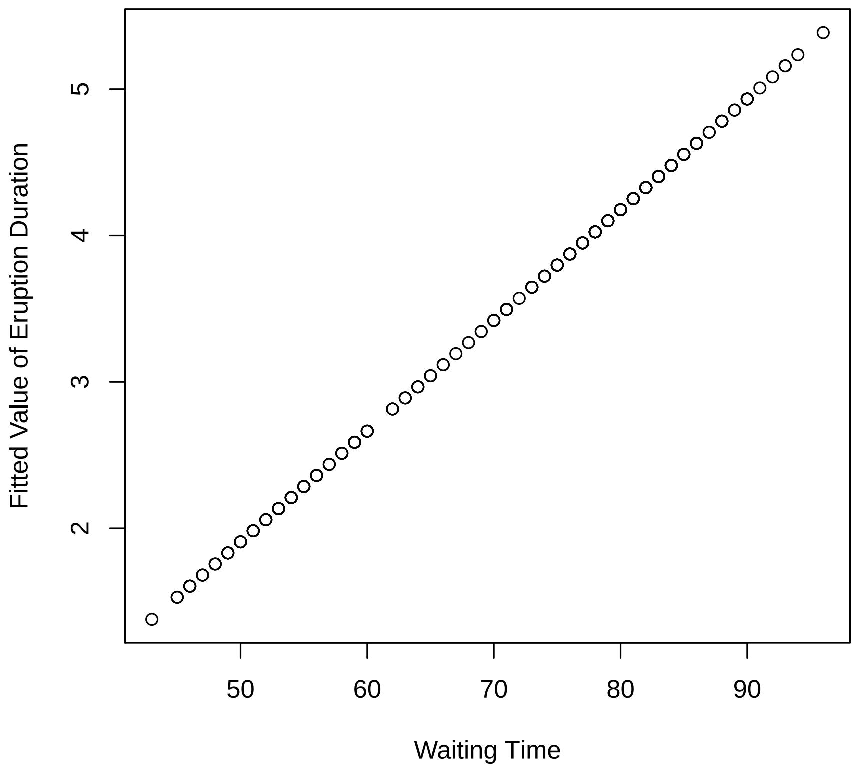
3.2.2 Examining the Residuals

The residual data of the simple linear regression model is the difference between the observed data of the dependent variable y and the fitted values $\hat{y}$.

$$Residual = y - \hat{y}$$

What if we plot the fitted values (you can get $\hat{y}$ by using the `fitted.values()` function in R) against the actual waiting times?

```
> plot(faithful$waiting, fitted.values(eruption.lm), xlab = "Waiting Time",
+     ylab = "Fitted Value of Eruption Duration")
```



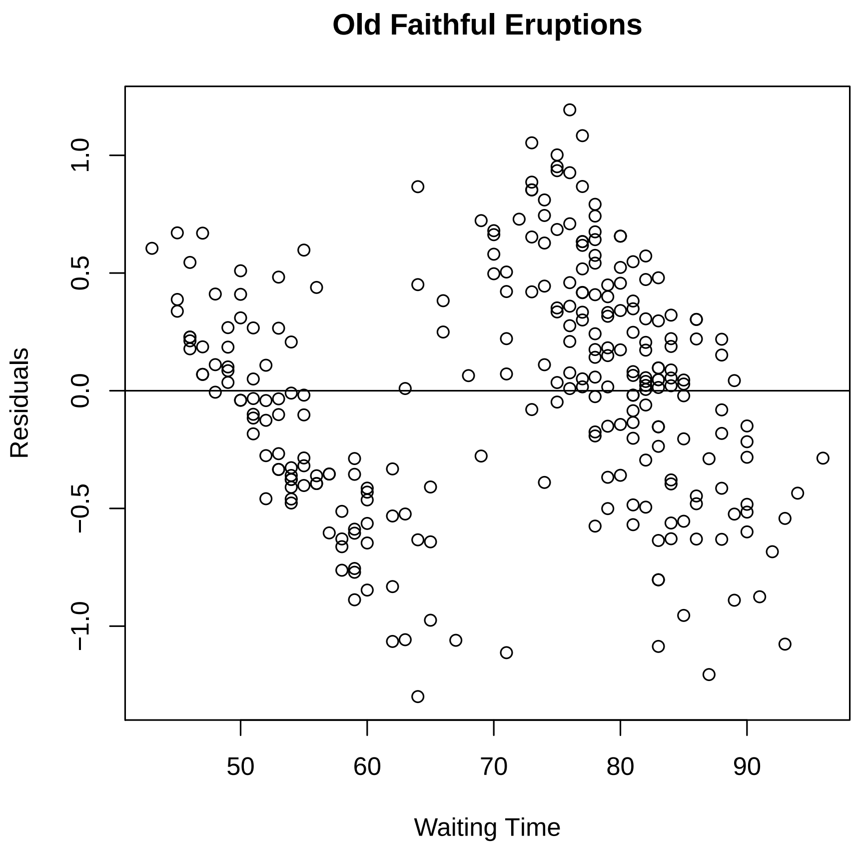
Now let's plot the residuals of the simple linear regression model of the data set *faithful* against the independent variable *waiting* (i.e. for each value of *waiting* what does the residual look like?).

Earlier, we applied the `lm` function to model variable *eruptions* as a function of *waiting*, saving the linear regression model in *eruption.lm*. Now we compute the residual with the `residuals` function.

```
> eruption.res = residuals(eruption.lm)
```

Plot the residuals against the observed values of the variable *waiting*.

```
> plot(faithful$waiting, eruption.res, ylab = "Residuals", xlab = "Waiting Time",
+     main = "Old Faithful Eruptions")
> abline(0, 0)
```



Do the residuals here only represent noise?

4 Questions?