Module 3 Lab

This lab guides you through the mechanics of simple linear regression. This includes fitting the model, exploring the returned fit, and obtaining confidence and prediction intervals. We use husband and wife data from *OpenIntro Statistics*, provided for you in the Module 3 Lab folder.

Simple Linear Regression

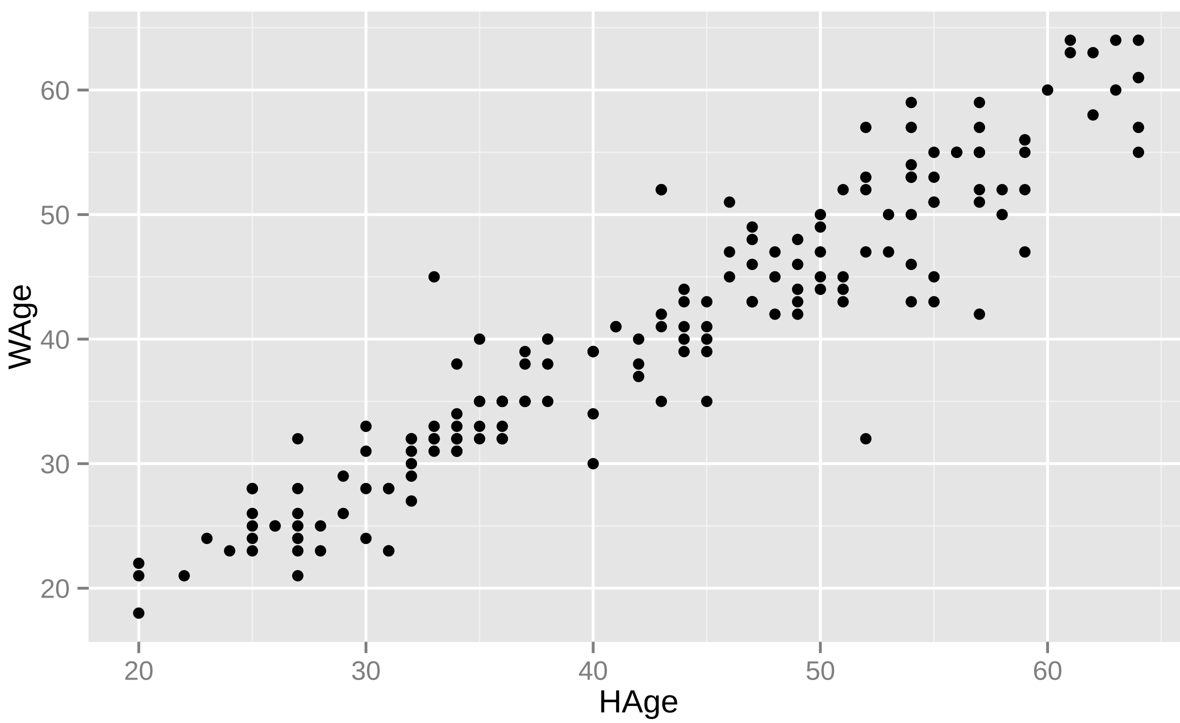
Load in the data and take a look at the first few rows.

```
HWData <- read.csv("Age.csv", row.name = 1) # Husband and Wife Data
head(HWData)
```

```
##   HAge HHght WAge WHght HAgeMar
## 1   49  1809   43  1590      25
## 2   25  1841   28  1560      19
## 3   40  1659   30  1620      38
## 4   52  1779   57  1540      26
## 5   58  1616   52  1420      30
## 6   32  1695   27  1660      23
```

The dataset has columns for the husband and wife age and height. Let's try to use the husband's age to predict his wife's age. We start with a scatterplot of the data.

```
library(ggplot2)
qplot(HAge, WAge, data = HWData)
```



The explanatory variable HAge is on the x-axis, and the response variable WAge is on the y-axis; this arrangement is customary. Also notice that qplot() removed the rows with missing data before plotting, which is helpful.

As far as the relationship between spousal ages, what do we see? There is a strong, positive linear correlation between a husband's age and his wife's age. Without further ado, let's fit a simple linear regression model.

```
fit <- lm(WAge ~ HAge, data = HWData) # Regress Wife Age against Husband Age
```

The function lm(), which stands for "linear model", regresses WAage on HAge. Notice that the structure for this formula in lm() is response ~ explanatory. We stored the model fit as fit, and you will notice there is no output. To get lm() to share, we need another line of code.

```
summary(fit)
```

```
##
## Call:
## lm(formula = WAge ~ HAge, data = HWData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.9586  -1.9897  -0.1035   1.8536  13.3550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.57401    1.15012   1.369    0.173
## HAge         0.91124    0.02585  35.249   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.951 on 168 degrees of freedom
##   (29 observations deleted due to missingness)
## Multiple R-squared:  0.8809, Adjusted R-squared:  0.8802
## F-statistic:  1243 on 1 and 168 DF,  p-value: < 2.2e-16
```

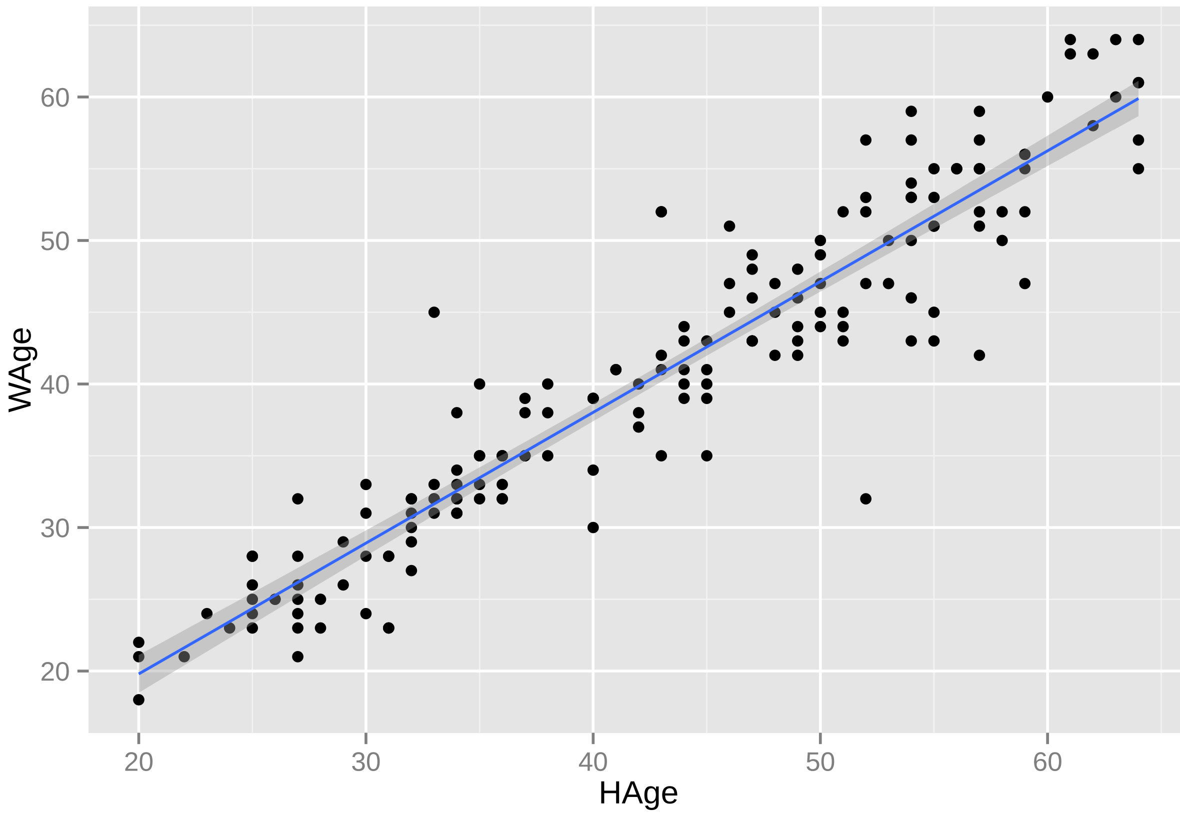
Starting at the top, we see the model that was fit inside lm(), followed by a five number summary for the residuals. Next the coefficient estimates are given in tabular form, along with their standard error, t-statistic, and associated p-value. Recall the model we fit,

$$WAge_i = \beta_0 + \beta_1 HAge_i + \epsilon_i,$$

where it is assumed that $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$. The R output above gives us $\hat{\beta}_0 = 1.57$, and $\hat{\beta}_1 = 0.91$. Next is the residual standard error $\hat{\sigma} = 3.96$. This is the estimate of the standard deviation of the ϵ_i's. In the same line, the output gives the degrees of freedom on which the estimate was based. The penultimate line of the R output gives two versions of R^2, which is described as the percent of variation explained by the model. The final line gives the F-statistic and p-value for the overall significance of the regression model, comparing the model we fit to a model that only includes the intercept.

We can replot our points, but this time with the regression line overlaid.

```
qplot(HAge, WAge, data = HWData) + geom_smooth(method = "lm")
```



You will notice the shaded bands around the regression line, which denote 95% confidence intervals for the mean wife age at each husband age. Later in this lab we learn how to calculate confidence intervals.

Residuals

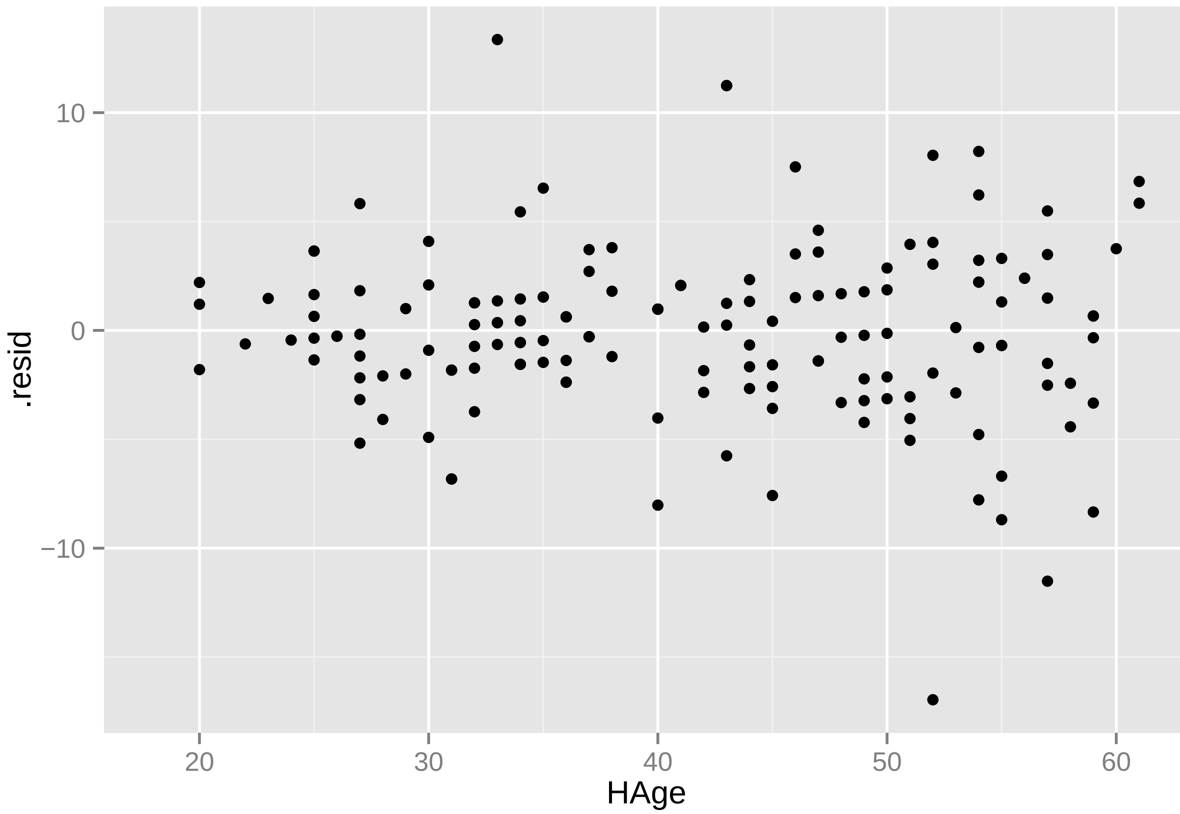
The stored regression object `fit` is actually a list with 13 elements. On a side note, `summary(fit)` is also a list, with 12 elements!

```
names(fit) # Show all 13 elements
```

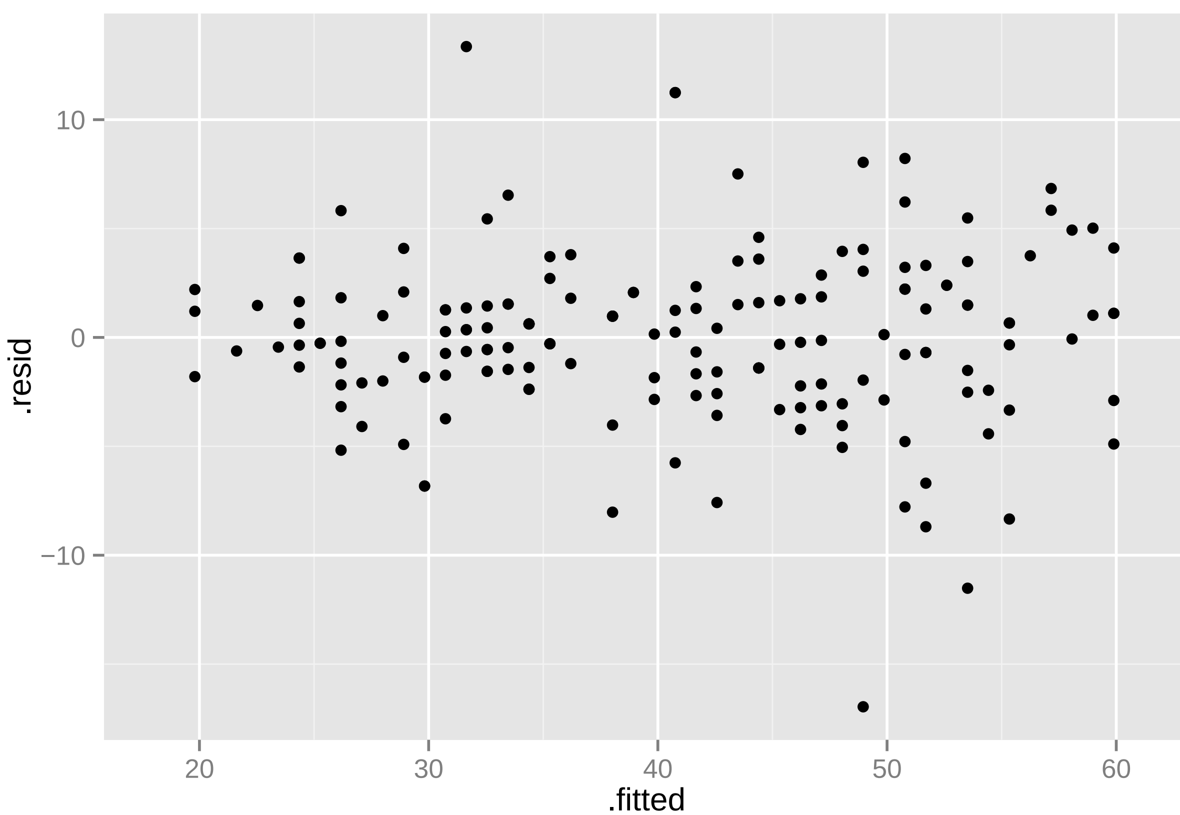
```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "na.action"     "xlevels"       "call"          "terms"
## [13] "model"
```

We can use the residuals to evaluate the validity of the model we fit, with regard to the model assumptions. If there is a reason to doubt the assumptions, it often reveals itself in non-random structure in a plot of residuals versus fitted values, or residuals versus explanatory variable values.

```
qplot(HAge, .resid, data = fit) + xlim(18, 61)
```




```
qplot(.fitted, .resid, data = fit) + xlim(18, 61)
```



Neither plot shows any systematic change in the variance across husband ages. More generally, there is no observable pattern or trend of any kind in the residuals. These plots do not raise any red flags, so we proceed to confidence and prediction intervals.

Prediction and Confidence Intervals

We may want confidence intervals for the estimated parameters. The function `confint()` will return these.

```r
confint(fit)
```

```
##                  2.5 %    97.5 %
## (Intercept) -0.6965354 3.8445513
## HAge         0.8602063 0.9622769
```

The output is self-explanatory. The default is a 95% confidence interval, but there is a `level = ?` argument to specify different confidence levels.

A second useful function is `predict()`, which calculates confidence and prediction intervals for specified values of the explanatory variable. For example, this code will produce point estimates and confidence intervals for the mean at all explanatory variable values in the original dataset.

```r
predict(fit, interval = "confidence")
```

What are the estimated mean wife ages for 30, 35, and 40-year-old husbands?

```r
new <- data.frame(HAge = c(30, 35, 40))
predict(fit, newdata = new, interval = "confidence")
```

On the other hand, what if we want to predict the wife age of one randomly selected husband for each of those ages?

```r
predict(fit, newdata = new, interval = "prediction")
```

Notice that in the first column the point estimates are the same. The prediction intervals, however, are much wider than the confidence intervals, as you can see in columns two and three.