Lab 6: Linear Models

ENVIRON 710: Applied Statistical Modeling*

In this lab, we will acquaint ourselves with simple linear models. Linear models describe the relationship between an explanatory variable and a response variable. We start with linear models with a single explanatory variable. Simple linear regression is the linear model used when both the response variable and the explanatory variable are continuous variables (i.e., real numbers with decimal places). The lab walks through all the steps of performing a linear regression from implementation to validation of the model.

But before we jump into linear models, we will first briefly discuss linear correlation. Correlation is *not* a linear model but is often used to examine the association between two variables which is useful when exploring our data.

The learning goals of the lab are to:

- Understand the basic formulae of linear models
- Understand correlation and how to implement it in R
- Practice implementing linear models (regression) in R
- Understand how to interpret the output for linear models
- Learn to read the diagnostic tests to evaluate the fit of models to data.

At the end of the lab, there are a few problems to answer. *Submit your answers in R Markdown to the class Sakai site under the Assignments folder.*

More functions in R

- `pairs()` - produces a matrix of scatter plots
- `ggpairs()` - produces a matrix of scatter plots using `ggplot2`
- `ggcorr()` - plots a correlation matrix with `ggplot2`
- `cor.test()` - test for association between paired samples, using Pearson's correlation coefficient, Kendall's τ or Spearman's ρ
- `lm()` - fits a linear model to the data; used for regression and ANOVA (when the independent variable is a factor)
- `source()` - calls a function or input from a named file, and is used to read the file and execute the commands in the file.

Linear Correlation

Linear correlation and linear regression are often confused, mostly because some bits of the math are similar. However, they are fundamentally different techniques.

If you have a set of pairs of values (call them x and y), you may ask if they are correlated. The two variables must both be observations for the correlation question to make sense. The underlying statistical model for correlation assumes that both x and y are normally distributed. In a correlation model, there isn't an independent and a dependent variable; both are equal and treated symmetrically. If you don't feel comfortable swapping x and y, you probably shouldn't be conducting a correlation analysis.

*Created by John Poulsen with edits from TAs.

The standard method is to compute the Pearson correlation coefficient. This method assumes a linear correlation between x and y. You could have strongly correlated data, but if the relationship is not linear the Pearson method will underestimate the degree of correlation, often significantly. Therefore, it is always a good idea to plot your data first. If you see a non-linear but monotonic relationship between x and y you may want to use the Spearman correlation - a non-parametric method - or transform your data so that the relationship becomes linear.

In correlation analysis, our principal interest is in the correlation coefficient, r. The correlation coefficient ranges from -1 to 1. A value of zero means that there is no correlation between x and y. A value of 1 means there is perfect correlation between them: when x goes up, y goes up in a perfectly linear fashion. A value of -1 is a perfect anti-correlation: when x goes up, y goes down in an exactly linear manner. In the Pearson method, r is calculated as:

$$r = \frac{cov_{xy}}{\sigma_x \sigma_y} = \frac{1}{n-1} \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

This is equivalent to taking the z-score of each variable, summing their cross products, and dividing by the degrees-of-freedom.

Note that x and y can be of different units of measure - this is because the equation converts the observations to z-scores so each value is the distance in units of SD's from the mean. Whenever both x and y are above or below their means, you get a positive contribution to r; when one is above and one is below you get a negative contribution. If the data are uncorrelated, these effects will tend to cancel each other out and the overall r will tend toward zero.

Correlation in R

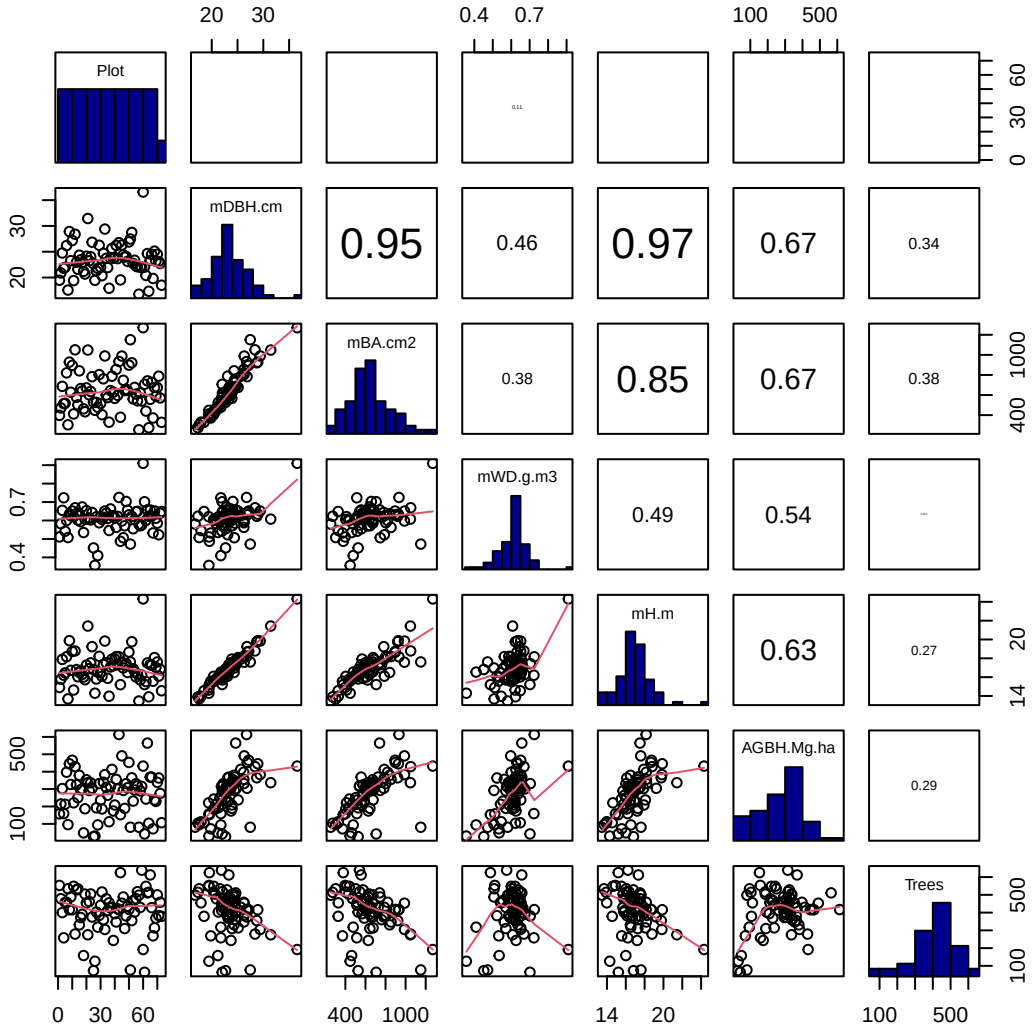
Download the dataset `TreePlots.csv`, which consists of data from 73 1-ha tree plots. The dataset includes a number of variables related to tree inventory, including: `Plot` (plot ID), `mDBH.cm` (mean DBH of trees in cm), `mBA.cm2` (mean Basal area in cm^2), `mWD.g.m3` (mean wood density in g m^{-3}), `mH.m` (mean height in m), `AGBH.Mg.ha` (aboveground biomass in Mg ha^{-1}), and `Trees` (number of trees).

To get a look at the data, we can dispense with all the histograms and scatter plots that we have been graphing one-by-one, and use the `pairs()` command to get a matrix of scatterplots of all the different variables. This provides bivariate scatterplots for all combinations of variables.

```
tdat  <-read.csv("TreePlots.csv", header=T)

 pairs(tdat)
```

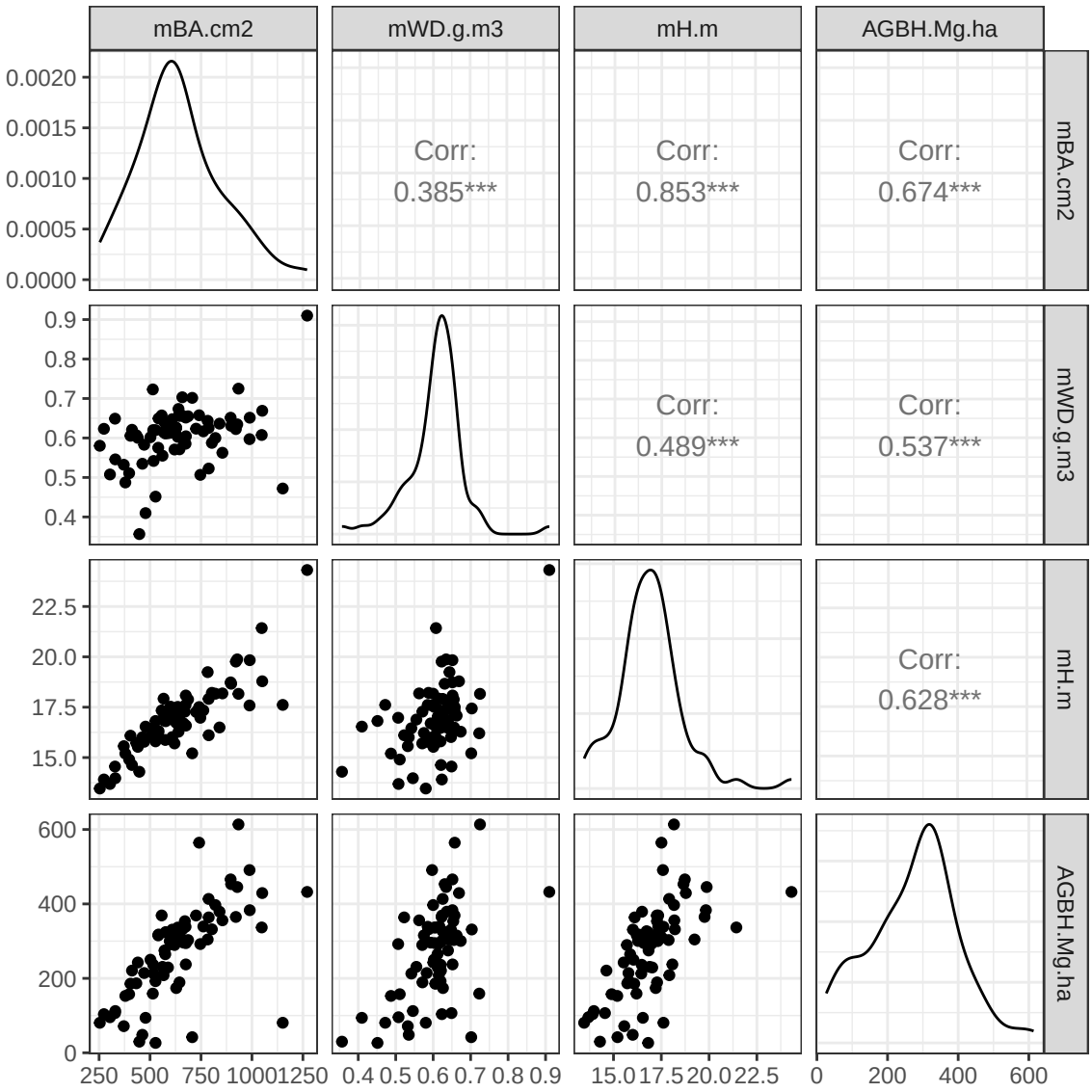
We can spruce up this graph and get even more information on the data, by using a script called `pair.fun.R`, which can be downloaded from Sakai.

```
source("pair.fun.R")

pairs(tdat, lower.panel=panel.smooth,
      upper.panel=panel.cor, diag.panel=panel.hist)
```

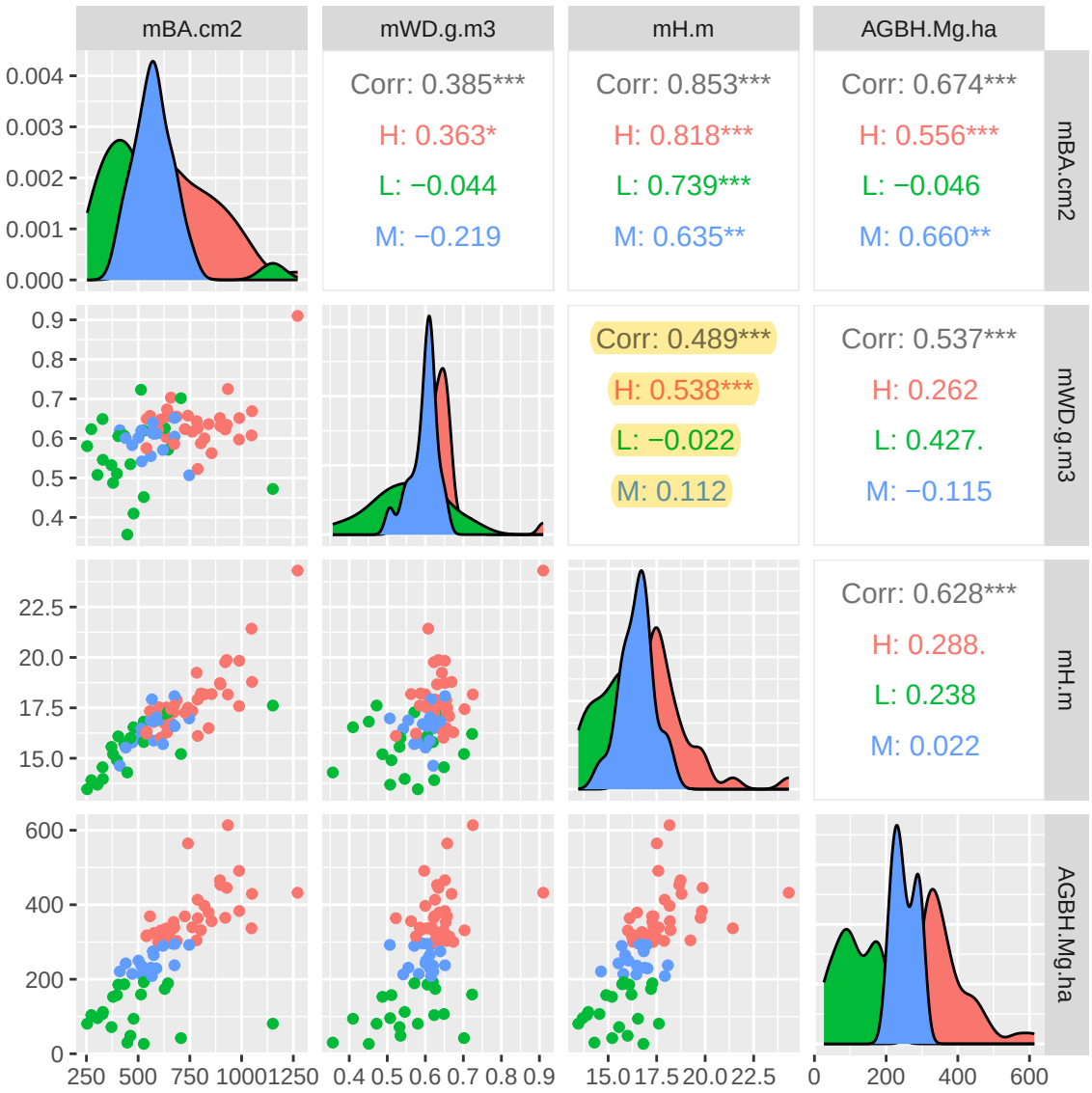


You can make a similar graph using `ggplot2` and the function `ggpairs`, but you need to install the `GGally` package. `ggpairs` is quite flexible and will make color graphs if a categorical variable is present in the dataframe. Below are two examples: the first just plots the continuous variables as above, and the second creates and adds a categorical variable to demonstrate plotting of categories in color.

```
require(ggplot2)
 require(GGally)

ggpairs(tdat,3:6) + theme_bw()
```




```
tdat$Level <- ifelse(tdat$AGBH.Mg.ha > 300, "H", "M")
 tdat$Level[tdat$AGBH.Mg.ha < 200] <- "L"
  tdat$Level <- as.factor(tdat$Level)

ggpairs(data = tdat, columns = c(3:6), mapping = ggplot2::aes(color = Level))
```



Study the plots and you should be able to get a good sense of the relationship between the different variables. These plots also provide the correlation coefficients for the relationship between each of the variables. We can verify this, by running a correlation on two of the variables, such as mean height and mean wood density.

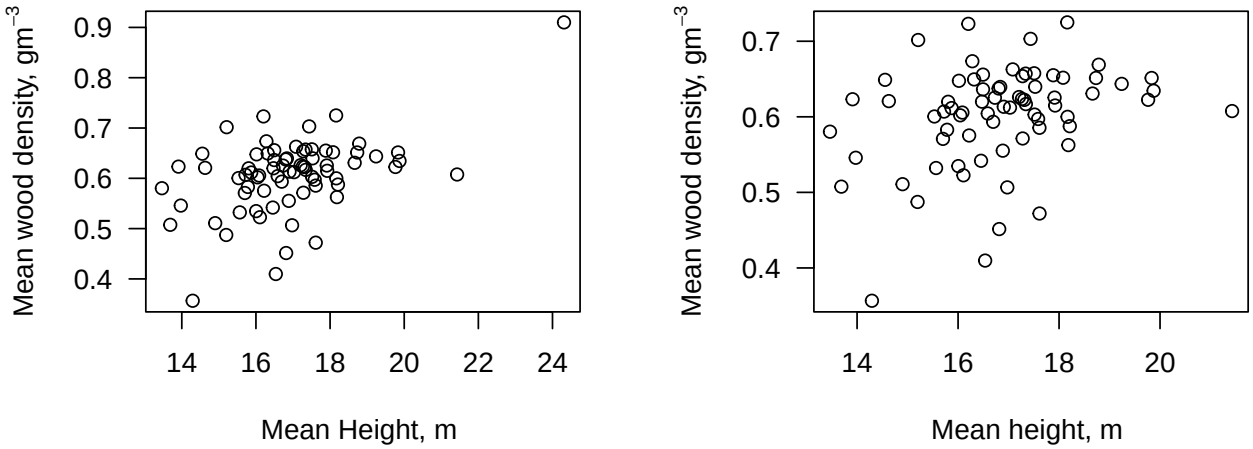
```
cor1 <- with(tdat, cor.test(mH.m, mWD.g.m3))
```

We could also calculate the r coefficient ourselves, by implementing the above equation in R.

```
tdat.short <- tdat[tdat$mH.m < 22,]
tdat.short <- tdat
 zx <- with(tdat.short, (mH.m - mean(mH.m))/(sd(mH.m)))
  zy <- with(tdat.short, (mWD.g.m3 - mean(mWD.g.m3))/(sd(mWD.g.m3)))

scp <- sum(zx * zy)
 r <- scp/(length(zx)-1)
```

Let's plot these two variables. In the first plot, the data point with the tallest mean tree height seems like it might be having a lot of influence on this correlation. In a second plot, we take it out and try again.

```
par(mar = c(5,5,6,2), mfrow = c(1,2))
with(tdat, plot(mH.m, mWD.g.m3, las = 1, xlab = "Mean Height, m ",
    ylab = expression(paste("Mean wood density, g", m^-3))))

with(tdat[tdat$mH.m< 22,], plot(mH.m, mWD.g.m3, las = 1,
                            xlab = "Mean height, m ",
                             ylab = expression(paste
                                    ("Mean wood density, g", m^-3))))
```



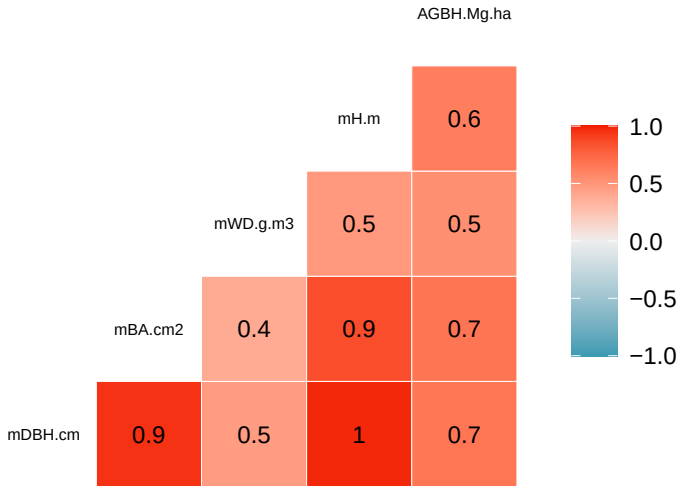

```
cor2 <- with(tdat[tdat$mH.m<22,], cor.test(mH.m, mWD.g.m3))
```

Compare correlations of the variables with and without that datapoint. What is the difference in correlation coefficients between the two analyses? Here we used correlation to check out how it works. Would we have been justified in using linear regression? Why or why not?

To Do

What does the output from `cor.test()` tell you? Can you interpret all the statistics? What is the null hypothesis for the test? Also, look up `?cor.test` and see what other options exist for correlations. Run a Spearman's non-parametric correlation and see whether the result is the same.

While `cor.test()` runs a correlation on two variables and provides tests of significance, you can find the correlation for an entire matrix using `cor()`. Note that we have to remove any non-numeric columns. The `ggcorr()` function plots a correlation matrix.

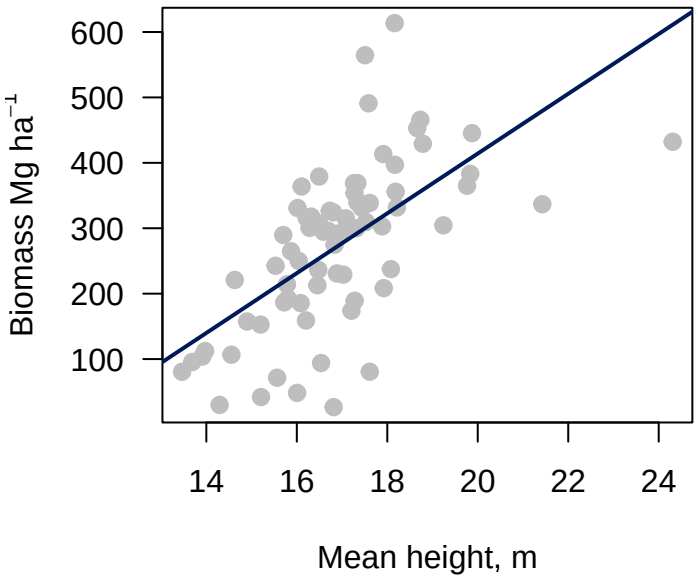


Linear Regression

Regression, or curve fitting, is a much richer framework than correlation. Correlation measures the degree to which the two variables are related, whereas regression is a method for describing the relationship between two variables. Regression allows us to *predict* new values for an outcome given measured values for an explanatory variable. Regression also allows us to *test a hypothesized model* of a system against data.

Now let's do a linear regression on tree height and aboveground biomass. Our hypothesis is that plots with greater mean height of trees have greater biomass. (You don't have to be a genius to figure that out, but let's proceed anyway.) Below, run the model, then test the $H_0 : mH.m = 0$, then plot the data.

```
lm1 <- lm(AGBH.Mg.ha ~ mH.m, data = tdat)
anova(lm1)

dukeblue <- "#001A57"

with(tdat, plot(mH.m, AGBH.Mg.ha, lwd = 2, las = 1, pch = 19,
     col =  "grey", cex = 1, xlab = "Mean height, m",
      ylab = expression(paste("Biomass Mg ", ha^-1))))

with(tdat, abline(lm(AGBH.Mg.ha ~ mH.m), col = dukeblue, lwd = 2))
```




```
summary(lm1)
```

Let's start at the very bottom of the results printout and look at the ANOVA printout. The small p-value tells us that our regression model explains a significant part of the variation in the data (more than the null model with just the intercept). It tests our alternative hypothesis (tree height is related to biomass; $H_0 : \beta_1 \neq 0$) against the null hypothesis (tree height has no effect on biomass; $H_0 : \beta_1 = 0$), and demonstrates that by incorporating tree height, $\beta_1 X_1$, we can explain a significant part of the variation in above ground biomass, Y_i.

$$H_0 : Y_i = \beta_0 + \varepsilon_i$$
$$H_a : Y_i = \beta_0 + \beta_1 X_1 + \varepsilon_i$$

If we had additional independent variables (multiple regression), then it would test:

$$H_0 : \beta_1 = \beta_2 = \ldots \beta_p = 0$$

$$H_a : \text{at least one of } \beta_1, \beta_2 \ldots \beta_p \neq 0$$

In a simple model like this, the ANOVA table does not provide much information different from the `lm()` output. However, ANOVA tables can be useful for comparing more sophisticated models. Remember, you can get the entire ANOVA table by with `summary.aov()`.

```
summary.aov(lm1)
```

Let's work through the rest of the information from `summary` piece-by-piece.

1. `Call` repeats the function call to the `lm()` function.
2. `Residuals` displays a five-number summary of the model residuals. This is the output obtained from applying the `quantile()` function to the residuals. The residuals are defined as $e_i = y_i - \hat{y}_i$ where y_i is the observed value and $\hat{y}_i$ is the predicted value (from the model) of the response variable for that observation.

   For example, take the case of Plot 1, where the observed biomass, `AGBH.Mg.ha`, is 157.45 and the observed mean height, `mH.m` is 14.90. Plug the height value into our regression formula:

$$\hat{y} = -499.926 + 45.704 \times 14.90 = 181.09$$

And, the residual would be $e_i = 181.09 - 157.45 = -23.64$. To get all the residuals for the model, use `residuals(lm1)` or `lm1$residuals`.

3. `Coefficients` contains the parameter estimates from the regression model. In this case, they are the y-intercept and slope of the regression line. The estimated model (rounded to 2 decimal places) is:

$$\hat{y}_i = -499.94 + 45.70 \times mH.m_i$$

We can use the equation to calculate the fitted values – the values for a response variable that have been predicted by a model fitted to a set of data.

```
y.fitted <- with(tdat, lm1$coefficient[1] + lm1$coefficient[2]*mH.m)
```

This should produce the same numbers as `lm1$fitted`. To check that they are the same, let's run a correlation between them:

```
cor.test(y.fitted, lm1$fitted)
```

What is your conclusion? (Hint: The numbers are perfectly correlated, meaning they are identical.)

Back to the regression model... Because the coefficient, `mH.m`, is positive the model predicts that aboveground biomass will increase as tree height increases. Every meter in additional height will lead to a 45.70 Mg ha^{-1} increase in biomass. The intercept in this model is not interpretable. Technically it represents the biomass of a plot when the average height of trees is 0. The model predicts a negative biomass when trees have a height of 0, which is nonsensical. In discussing the intercept we are extrapolating beyond the range of data that are possible, one of the no-no's in regression analysis.

The rest of the display includes the standard deviation of the estimates, and the t statistic and corresponding p-value for the null hypothesis that the true value of each coefficient is 0.

The individual t-tests are variable-added-last tests. They test whether the coefficient of the given variable is significantly different from 0, in the context of a model that already contains the rest of the independent variables. These tests are not tests of the importance of a variable by itself, but only in the context of the rest of the model. More about this when we discuss whether to include a variable in a model or not.

4. `Residual Standard Error`, $\hat{\sigma}$, is the square root of the sum of the squared residuals divided by their degrees of freedom, where n is the number of observations and p is the number of estimated regression parameters ($73 - 2 = 71$, in this example).

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n-p}}$$

When the residual standard error is 0, then the model fits the data perfectly (likely due to overfitting). If the residual standard error cannot be shown to be significantly different from the variability in the null model, then there is little evidence that the linear model has any predictive ability.

5. `Multiple R-squared` is the coefficient of determination: the proportion of the variation in the response that is explained by the regression. Here 39.5% of the variation in aboveground biomass is explained by its linear relationship to mean tree height. R^2 compares the amount of unexplained variation before and after a regression model is fit. The before variation is the sum of squares total (SST).

The adjusted R-square introduces a penalty term for each regressor (independent variable) in the R^2 calculation. Without this adjustment, R^2 will usually increase (at least a little) with each additional regressor; and thus we will always end up choosing the most complicated model as the best model. The penalty in the adjusted R^2 is chosen to yield an unbiased estimate of the population R^2 and works along the same principal as the AIC, which we will talk about later.

Reporting results from linear regression

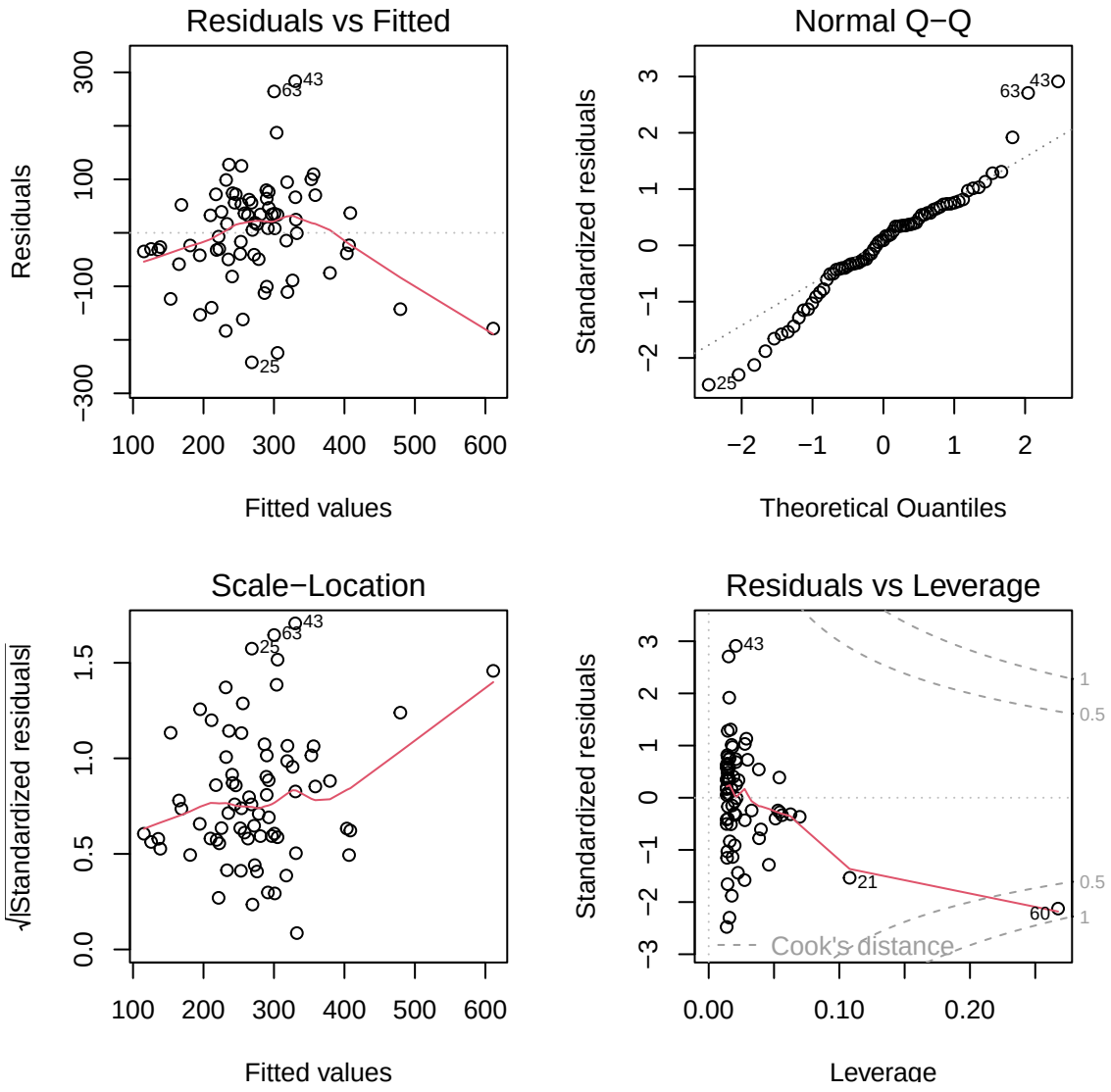
For the above example, we might say: average tree height in forest plots significantly increases plot-level biomass, with a 1 m increase in average tree height resulting in a 45.70 Mg ha^{-1} increase in plot biomass ($R^2 = 0.386$, $F_{1,71} = 46.34$, $p < 0.001$).

In some cases, you may want to present the entire equation: average forest biomass is significantly related to average tree height (biomass $= -499.9 + 45.70 \times$ height, $R^2 = 0.386$, $F_{1,71} = 46.34$, $p < 0.001$).

Checking the fit of the model

When conducting any statistical analysis it is important to evaluate (i) how well the model fits the data; and, (ii) that the data meet the assumptions of the model. There are numerous ways to do this and a variety of statistical tests to evaluate deviations from model assumptions. Generally, we examine diagnostic plots after running regression models, as we did for ANOVA. See the below description of the diagnostic tests.

```
par(mfrow=c(2,2), mar = c(3.8, 4, 3, 2))

plot(lm1)
```



Plotting the model provides four diagnostic plots. The first is a plot of the residuals (distance of the data points from the fitted regression line) versus the fitted data. Points should be randomly scattered around the centerline. Any pattern indicates either violation of linearity or homoscedasticity.

The second plot is a q-q plot, which we have already used to evaluate the normality of a variable. Significant departures from the line suggest violations of normality. If the pattern were S-shaped or banana shaped, we would need a different model. You could perform a Shapiro-Wilk test of normality with the `shapiro.test()` function, but be careful...

Against better judgment, in the past we have used the `shapiro.test()` to assess normality. Remember that no test will show that your data have a normal distribution. Normality statistics show when your data are sufficiently inconsistent with a normal distribution that you would reject the null hypothesis of "no difference from a normal distribution". However, when the sample size is small, even big departures from normality are not detected, and when the sample size is large, even the smallest deviation from normality will lead to a rejected null. In other words, if we have enough data to fail a normality test, we always will because real-world data won't be clean enough. See (http://www.r-bloggers.com/normality-and-testing-for-normality/) for an example with simulated data. So, where does that leave us? Explore your data for large deviations from normality and make sure to assess heteroscedasticity and outliers. But, don't get hung up on whether your data are normally distributed or not. As the author of the above link suggests: "When evaluating and summarizing data, rely mainly on your brain and use statistics to catch really big errors in judgment."

The third plot is a plot of standardized residuals versus the fitted values. It repeats the first plot, but on a different scale. It shows the square root of the standardized residuals (where all the residuals are positive). If there was a problem, the points would be distributed inside a triangular shape, with the scatter of the residuals increasing as the fitted values increase.

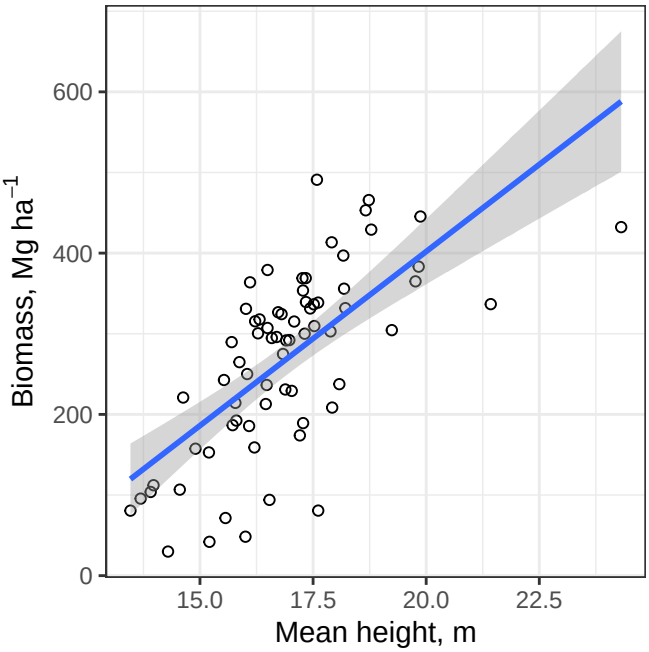
The fourth plot is a residuals-leverage plots that shows Cook's distance for each of the observed values. Cook's distance measures relative change in the coefficients as each replicate is deleted. The point is to highlight those y_i (response) values that have the biggest effect on parameter estimates. The idea is to verify that no single data point is so influential that leaving it out changes the structure of the model. The potential trouble points are labeled.

In the diagnostic plots for our model, both the Scale-Location and Residuals vs. Leverage plots show points scattered away from the center, suggesting that some points have excessive leverage. Another pattern is that points 25, 43, and 63 stick out in nearly every plot. This warns us that something could be odd with those observations. We might want to redo the analysis without those points (tree plots) and see if our inference changes.

```
lm2 <- with(tdat[-c(25,43,63), ], lm(AGBH.Mg.ha ~ mH.m))
```

Taking out these observations actually improves our R^2 and doesn't change the estimate for the effect of mean tree height, `mH.m`, much at all. We can plot the regression line, this time getting fancy with `ggplot2`.

```
require(ggplot2)
 tdat1 <- tdat[-c(25,43,63), ]

  ggplot(tdat1, aes(x = mH.m, y = AGBH.Mg.ha)) +
   geom_point(shape = 1) +
    geom_smooth(method = lm) +
     xlab("Mean height, m") +
     ylab(expression(paste("Biomass, Mg ", ha^-1))) +
    theme_bw()
```



`ggplot2` offers a lot of options and makes attractive graphs, but it is complicated for new users so feel free

to stick with the standard plotting procedures that we have used up to now. Note that using `geom_smooth` produces a 95% confidence region. This can be suppressed by revising the above to read: `geom_smooth(method = lm, se = F)`.

Problems

1 & 2. Your assignment is to conduct two different linear regressions on the `TreePlots_Lab6.csv` data. *Note that this is a different dataset than was used above!!!* Model (1) mean tree diameter (`mDBH.cm`) versus plot biomass (`AGBH.Mg.ha`), and (2) mean height (`mH.m`) versus mean wood density (`mWD.g.m3`). In the first model, biomass should be your dependent variable. In the second regression, mean height should be your dependent variable.

> See my above comment about robustness of linear models to deviations from normality, and don't sweat departure from normality too much for this problem.
>
> For each regression, write a 1-page description of your analysis, results, and inference. Each write-up should include the following information:

- Null and alternative hypotheses of your tests
- Results of your statistical test, interpreting the results in 2-3 sentences that include the appropriate reporting of the statistics
- An interpretation of the regression model (equation) from each analysis (e.g. how does plot biomass and mean height vary with different levels of tree diameter and wood density)
- A description of how you checked the assumptions of your model
- An interpretation of diagnostic figures
- A scatter graph showing the data and the best-fit regression line.

3. Also model the relationship between brain weight, kg, and body weight, kg, of different animals. Test the alternative hypothesis that brain weight increases with body weight. If the data show large deviations from normality, then use an appropriate transformation. Complete all the bullet points in problem 1 for this problem.