Homework #4, Due Friday, Feb. 4

A. The probability model behind regression says that to determine the response of a "system", first set the predictor to a value x. Then the response will be (mean + error) where mean = a + bx and error = (random number, normally distributed, mean 0, sd =something.) In this exercise, you'll generate fake data that follows this model and do some exploring.

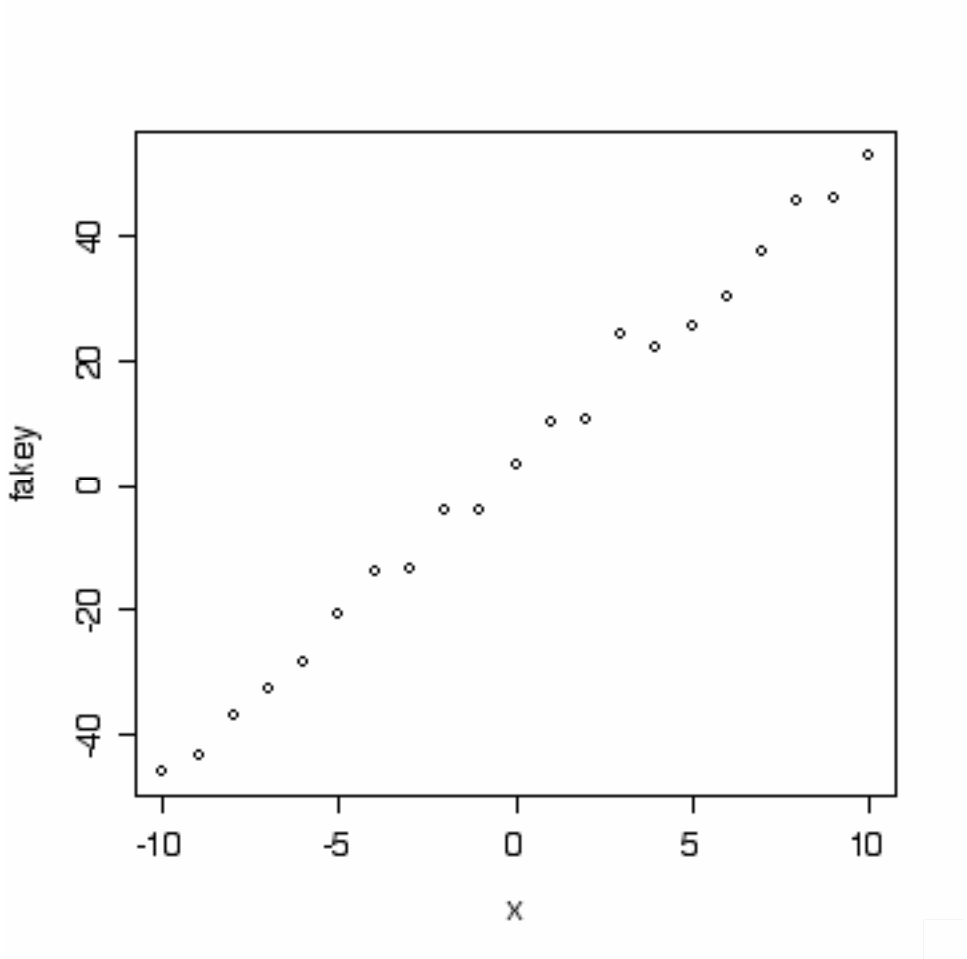
i) Write a function that creates fake data according to a simple regression model. The inputs to the model should be intercept, slope, standard deviation, x. The predictor values, x, should be a vector of values, and the response fo the function should be y: a vector of "observations" for the x. So you should be able to do something like this:

x <- -10:10
y <- yourfunctionhere(x, 3, 5, 2)
and y will be otuput for the model y = 3 + 5x + e where e is N(0,2).

Show me your function, and generate some data using the definition of x (-10, -9, ...9, 10) shown above.
Plot x vs. y.

```
makedata <- function(x, intercept, slope, sigma){
      deterministic <- intercept + slope*x
      randomerror <- rnorm(length(x), 0,sigma)
      deterministic+randomerror}
```

```
> x <- -10:10
> fakey <- makedata(x,3,5,2)
> plot(x,fakey)
```



The function could have been made much "cleaner":

```
myfunc <- function(x,int,slope, sigma){
        int + slope*x + rnorm(length(x),0,sigma)}
```

Will also work. Also, this version is pretty general (meaning it allows the user to control all of the parameters. From the wording of the HW problem, which asked for you to generate data using only one set of values, you could also have done something like;

```
myfunc <- function(){
        x <- -10:10
        3+5*x + rnorm(21,0,2)}
```

And then

```
fakey <- myfunc
```

would have worked. But to make a plot, you'd have to type

```
x <- -10:10
fakey <- myfunc
```

plot(x,fakey)

or you could type
plot(-10:10, myfunc())

ii) OPTIONAL: What is the population standard deviation of the y's? (Note: I'm not asking you to calculate it from the data -- that would be the sample standard deviation. I'm asking you to determine what the model says it should be.) Compare this to the sample standard deviation of the y's. (This is a little harder than I had intended. It requires that you know the population standard deviation of the x's. Feel free to think about it. In case you're wondering, the population sd of the x's is 5.91)

This was made optional (a bit too late for some -- sorry) because it's actually more complicated than I had realized when typing it, given what we covered in class. But here's the idea. The population standard deviation of y measures the variability in the vertical direction of all of the y values. The sample standard deviation of the y's (sd(fakey), which equals 30.84451 for my generated data) is an estimate of the population standard deviation.

The variability of Y comes from two sources: the variation in the random errors (which are the deviations above and below the regression line) and the spread of the x values. We haven't talked about x as a random variable, but intuitively, at least, you should see that the spread of the x's will affect the spread of the y's. Afterall, if we had only produced generated data for x values between -1 and 1, there would have been much less variation in the y values.

For those who remember their random variable "calculus":
if X and W are any two independent random variables, and a and b are any two constants, then
Var(aX + bW) = a^2 Var(X) + b^2 Var(W).
and Var(a) = 0.

According to the linear model:
Y = 3 + 5X + e
Var(Y) = Var(3) + 5^2 Var(X) + Var(e)

= 0 + 25*VarX + 4

VarX is the population variation of the values -10, -9, ..., 9,10.

```
> var(x)
[1] 38.5
```

So VarY = 25*38.5 + 4 = 966.5 and so the SD of Y is

```
> sqrt(966.5)
[1] 31.08858
```

iii) The model says that the RSS should be approximately equal to 4(n-2) . Show why. Then compare this to the estimate from the LS line.*

OK. I admit. This was a poorly worded question. Here's what I was thinking:

We estimate the standard deviation of the errors by sigma-hat^2 = RSS/(n-2) and so RSS = sigma-hat^2 * (n-2).

You can see these is the case by fitting a linear model to your generated data, finding the residuals, squaring them, and adding them up. This will give you the left hand side of this equation.

```
> fake.lm <- lm(fakey~x)
> fake.resid <- residuals(fake.lm)
> sum(fake.resid^2)
[1] 101.8033
```

You can get the estimated standard deviation of the errors from the summary command:

```
> summary(fake.lm)

Call:
lm(formula = fakey ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7371 -1.6772 -0.3725  1.0161  6.1574
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.05278    0.50512   6.044 8.19e-06 ***
x            4.95773    0.08342  59.432  < 2e-16 ***
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 `
' 1

Residual standard error: 2.315 on 19 degrees of freedom
Multiple R-Squared: 0.9946,   Adjusted R-squared: 0.9944
F-statistic:  3532 on 1 and 19 DF,  p-value: < 2.2e-16
```

It's called "residual standard error" and it's equal to 2.315. (It should be about 2. The population standard deviation that this residual standard error estimates is the value you typed into your function to generate the random errors.)

And n = 21, so you can check that the two sides are right.

The reason I asked for what the "model" says RSS should be is that in the population, the residuals standard error is 2, and so if we saw all values in the infinitely large population, RSS/(n-2) should equal 2^2 = 4.

iv) Check the residual plots to check for linearity and normality. Of course you know these should be good because you made them that way. But the idea is to get some experience seeing what "good" plots look like.

To check for linearity, you can either plot the residuals against the x values, or against the predicted values (also called the "fitted values" also called the "y-hats"). The following commands will produce the same plot:

```
> fake.resid <- residuals(fake.lm)
> plot(x,fake.resid)
OR
> fitted <- fake.lm$fitted.values
> plot(fake.resid,fitted)
```

OR

```
> fitted <- predict.lm(fake.lm)
> plot(fake.resid,fitted)
```

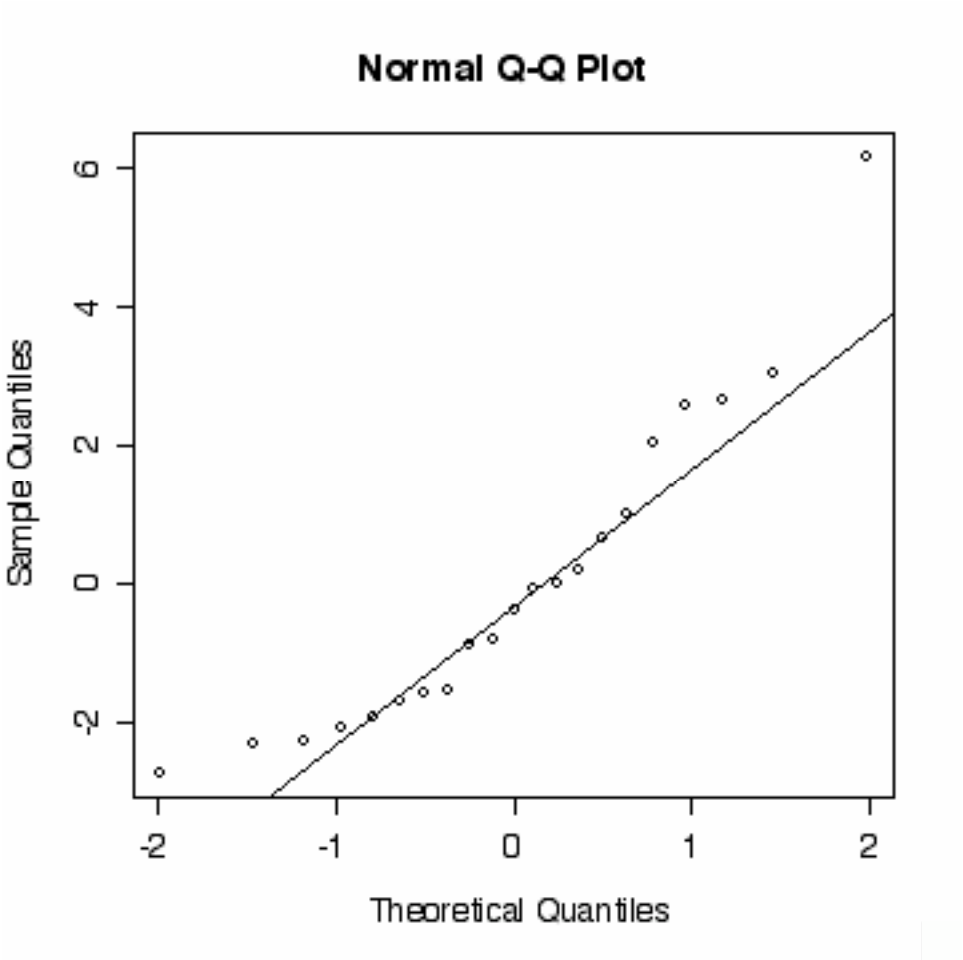
OR

```
> plot(fake.lm)
Hit <Return> to see next plot:
```

If the data are consistent with the linear model, there should be no pattern to this residual plot. The residuals should be more-or-less evenly distributed in a band about the line resid=0 (and the width of this band should be more-or-less constant).

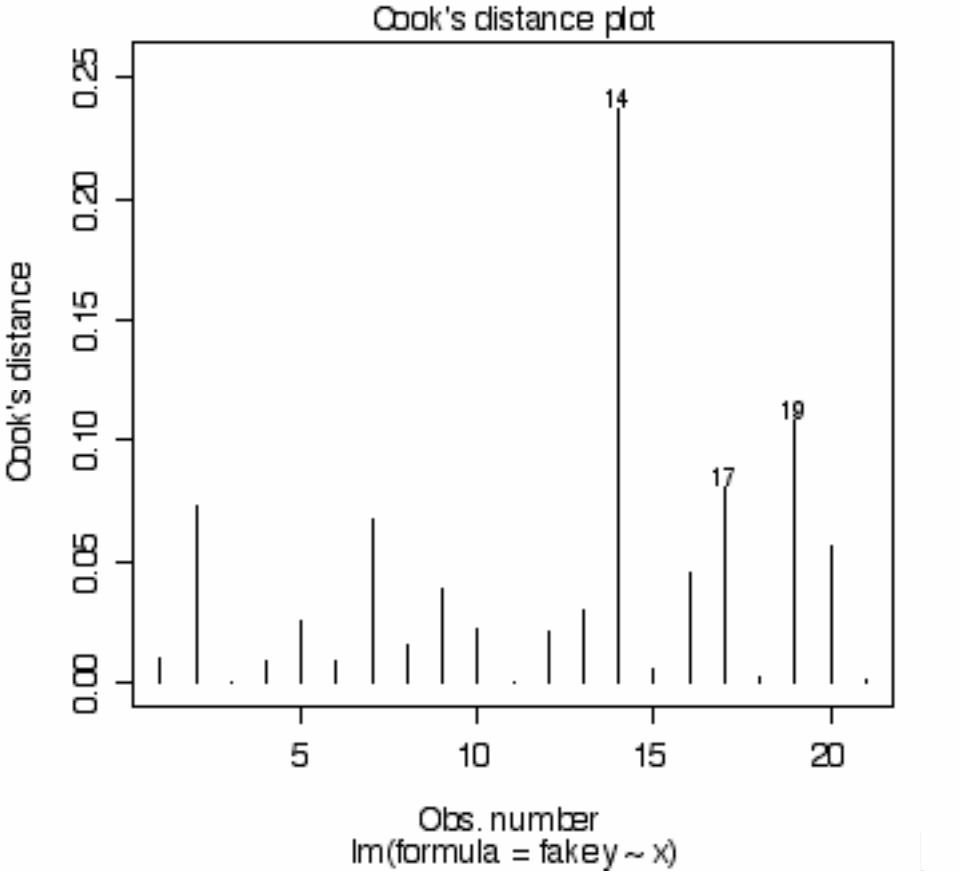
To check linearity, the best plot is the qqnorm:

```
> qqnorm(fake.resid)
> qqline(fake.resid)
```



If the residuals are normally distributed, this plot should be a fairly straight line. Note that we know for a fact that these are normally distributed because we made them that way. I got a little unlucky with my random sample, but this is not all that unusual for a small sample size.

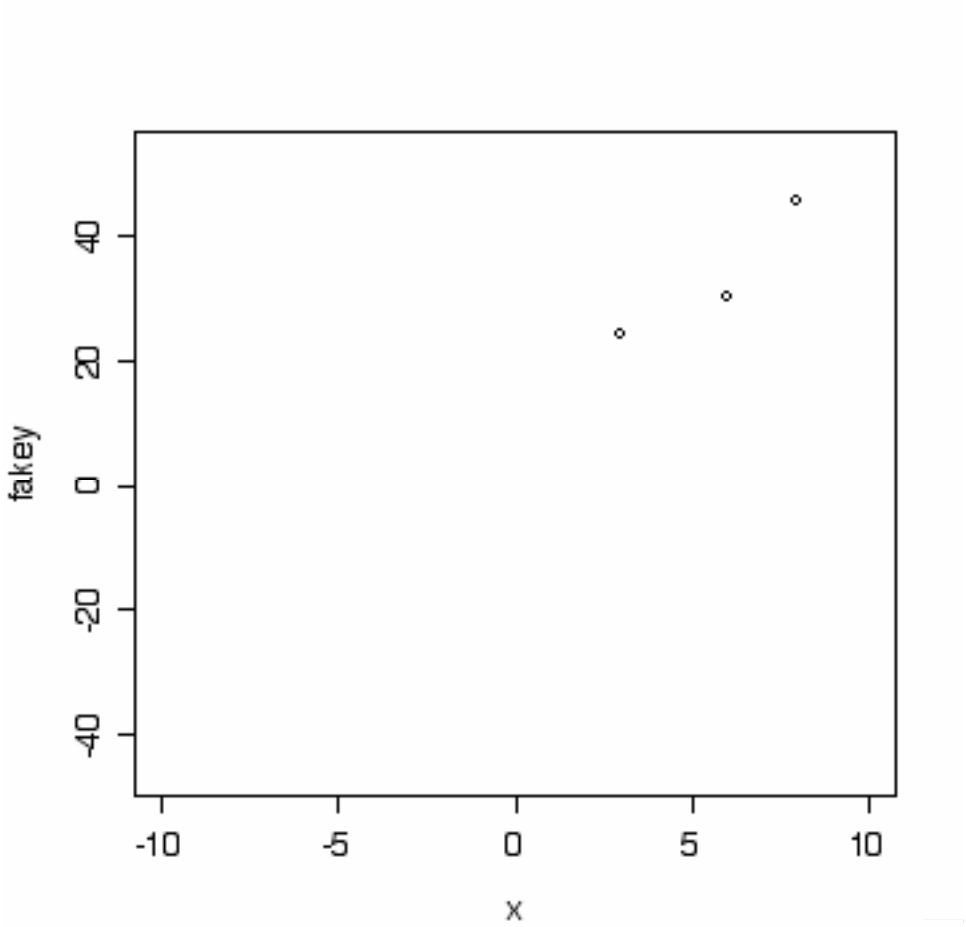
v) Which values of x have the largest Cook's distance for your fitted model? (If "myoutput" is the output of a call to the lm command, then plot(myoutput) will give you a series of four plots (just hit return to see the next one). The last plot shows Cook distances.)



For me it looks like observation 14, 17, and 19 had large influence (which means that if I were to fit a new line without one of those points, the new line would be fairly different from the old).

Which points are these? Here's a plot with only those points:

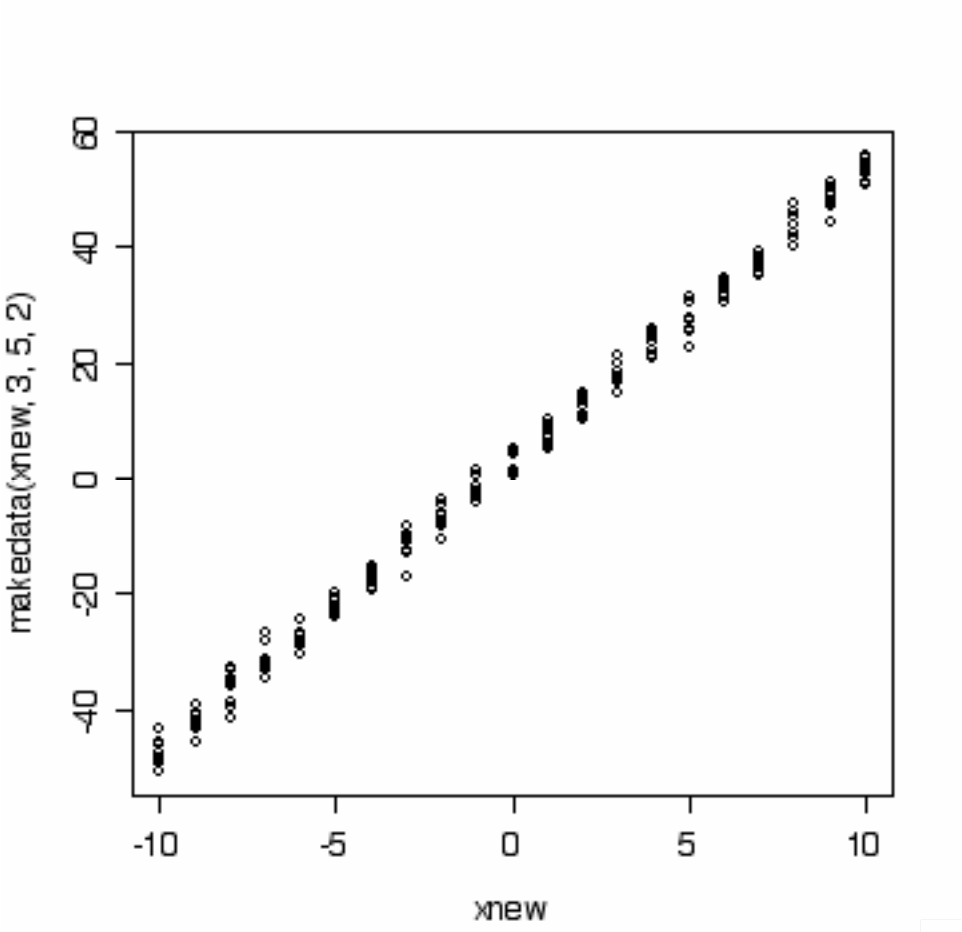
```
> plot(x,fakey,type="n")
> points(x[c(14,17,19)],fakey[c(14,17,19)])
```



>

You can see that they're all three points out at the extremes, which is not at all an unusual position for an influential point.

vi) Repeat (ii)-(v) using xnew <- rep(x,10). Make sure to plot (xnew, ynew).

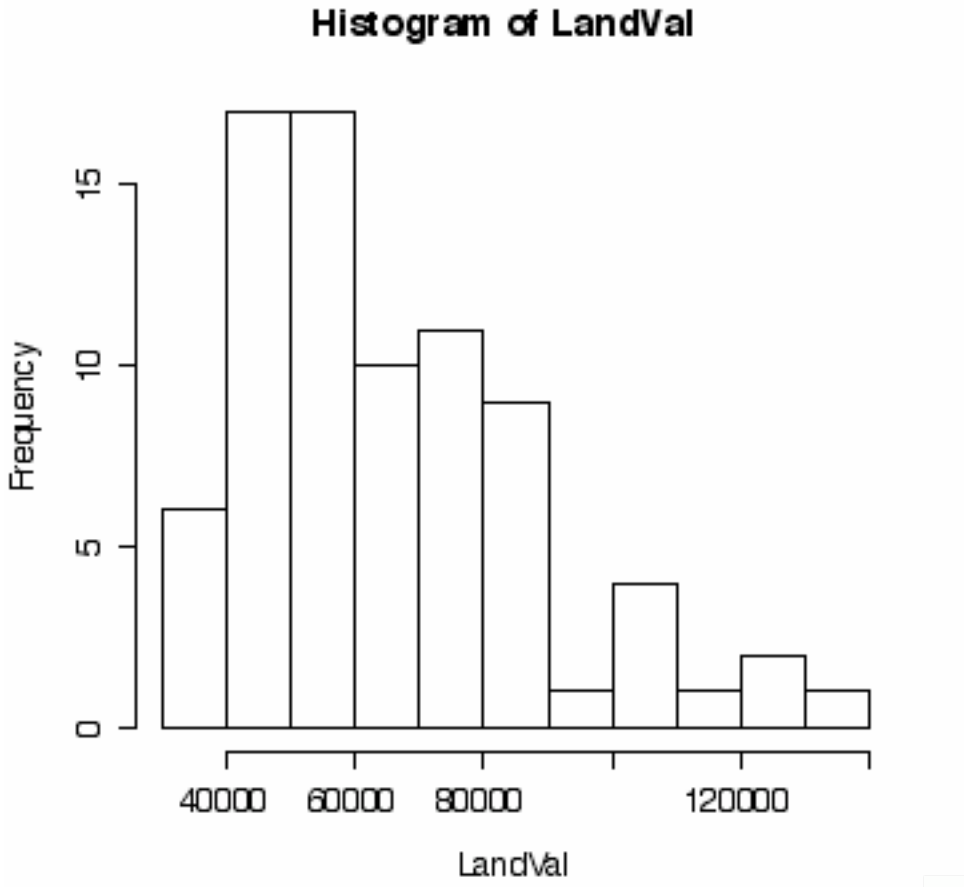


B. What makes one piece of real estate property more or less valuable than another? The land.txt datafile (tab-delimited, text) records values on several variables that could affect the cost of a several properties. This dataset consists of 11 variables. The variable named "LandVal" gives the value of the land alone – without consideration of the structure on it. "TotalVal" is the value of the property as a whole. "Acreage" is the size of the lot and the other ten variables describe key features of the buildings on the parcels of land. "Height" is the number of floors, "fl1Area" is the square footage of the first floor, "Rooms", "Bedrooms", "FullBath", "HalfBath", "Fireplce", and "Garage" are all exactly what they seem to be.

i) Make a histogram of LandVal. What does this say about the value of these parcels of land.

```
> land.table <- read.table("land.txt", header=T, sep="\t")
> names(land.table)
```

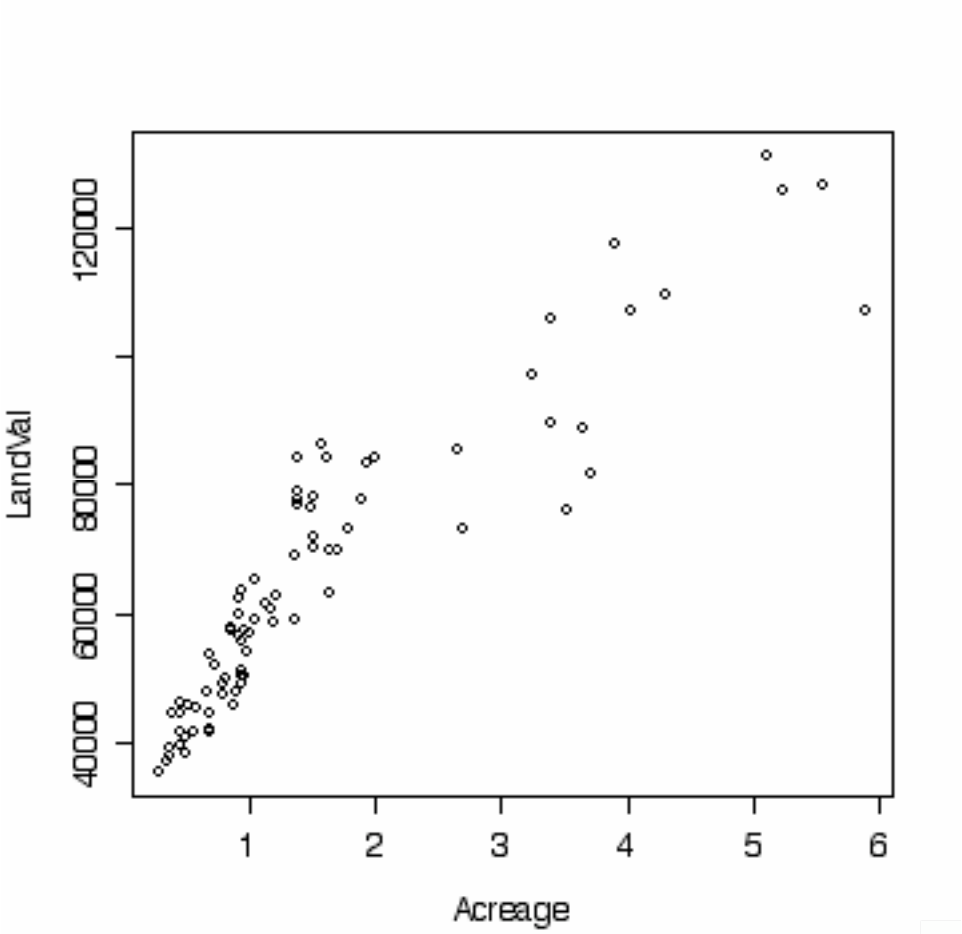
```
 [1] "LandVal"  "TotalVal" "Acreage"  "Height"   "fl1Area"
"Rooms"    "Bedrooms"
 [8] "FullBath" "HalfBath" "Fireplce" "Garage."
> attach(land.table)
> hist(LandVal)
```



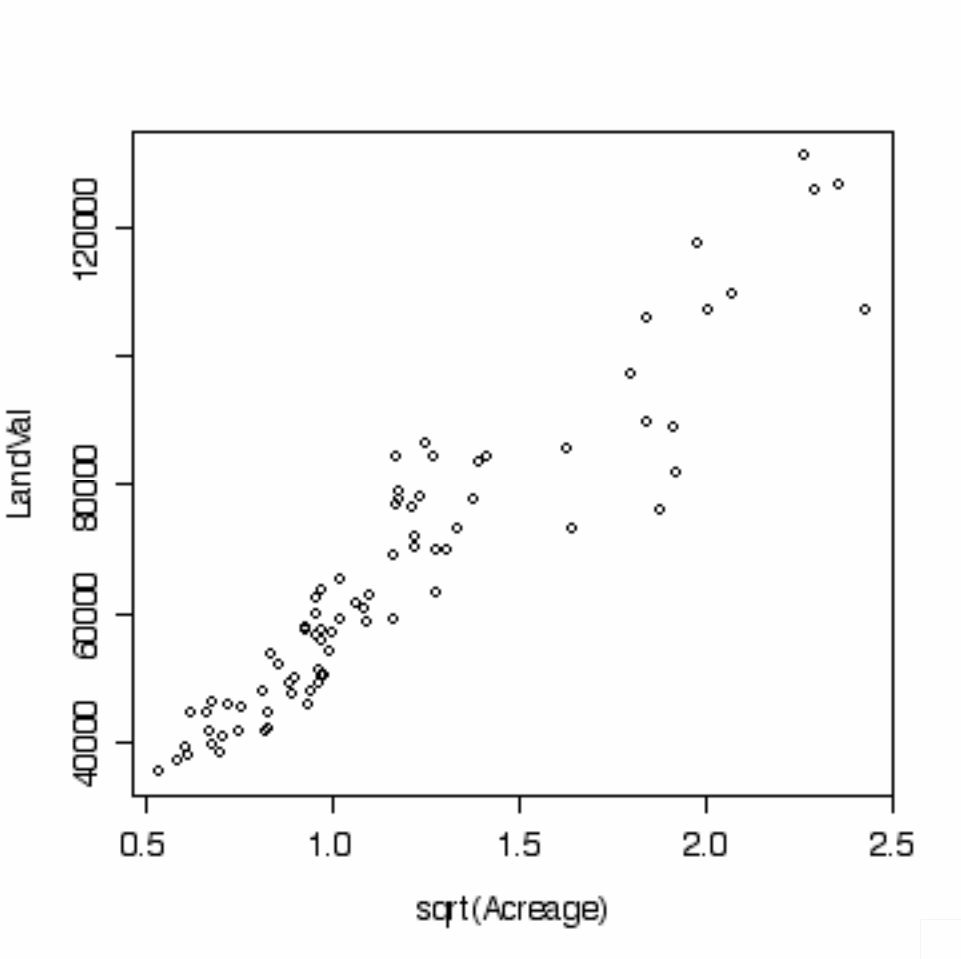
These properties are valued between about 40K and 130K (those were the days). The distribution is right-skewed, with the bulk of the properties under 80K.

ii) Perform a regression to use Acreage to predict LandVal. Your regression should include

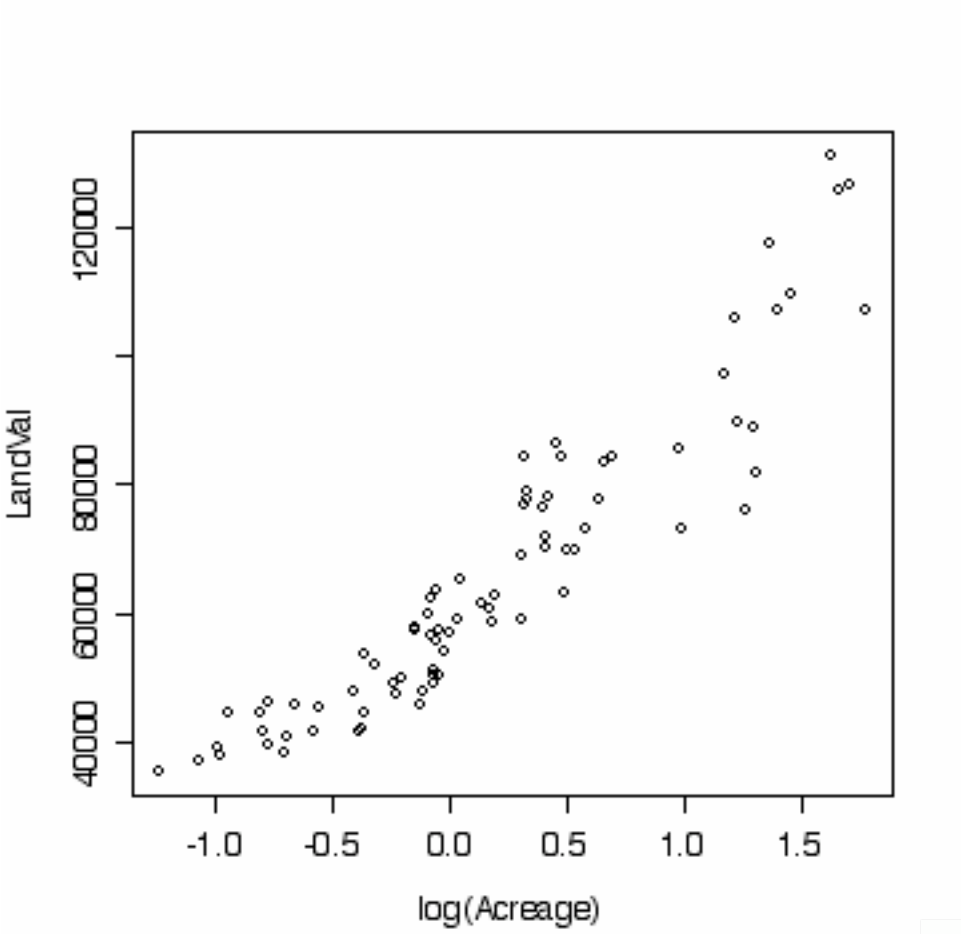
a) an initial scatterplot
b) a final model
c) discussion of any needed transformations
d) discussion of the validity of any assumptions
e) An interpretation of what the model tells us about this relationship and how well the model fits.



An initial scatterplot shows a potential non-linear relationship between these variables. Probably we'll need to transform one or both of the variables to make the relationship more linear. Here's the plot against the square-root of Acreage:



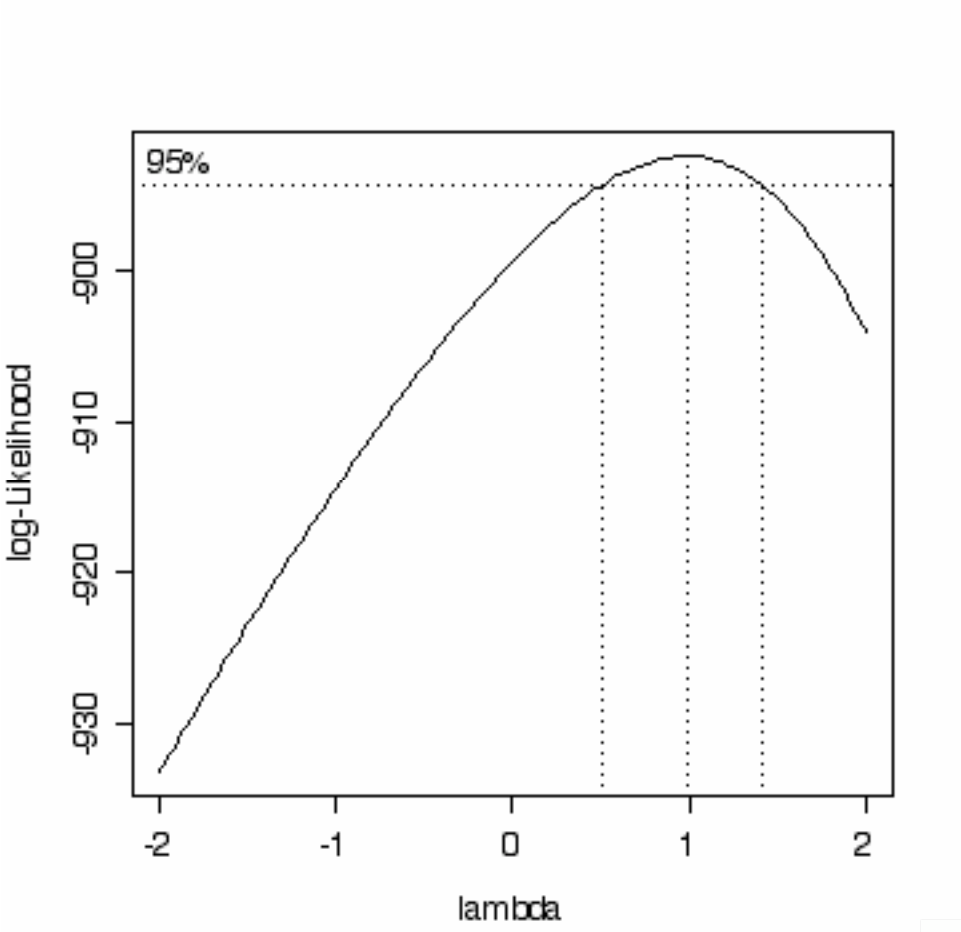
And here's a plot of the log of acreage:



To my eye, the square-root looks better. The log seems to have a slight bend in it.

We get some confirmation from looking at the boxcox function. This function, remember, recommends a parameter which we'll call L. Roughly speaking, L is the power that you should raise x to (x^L) in order to get the most linear transformation.

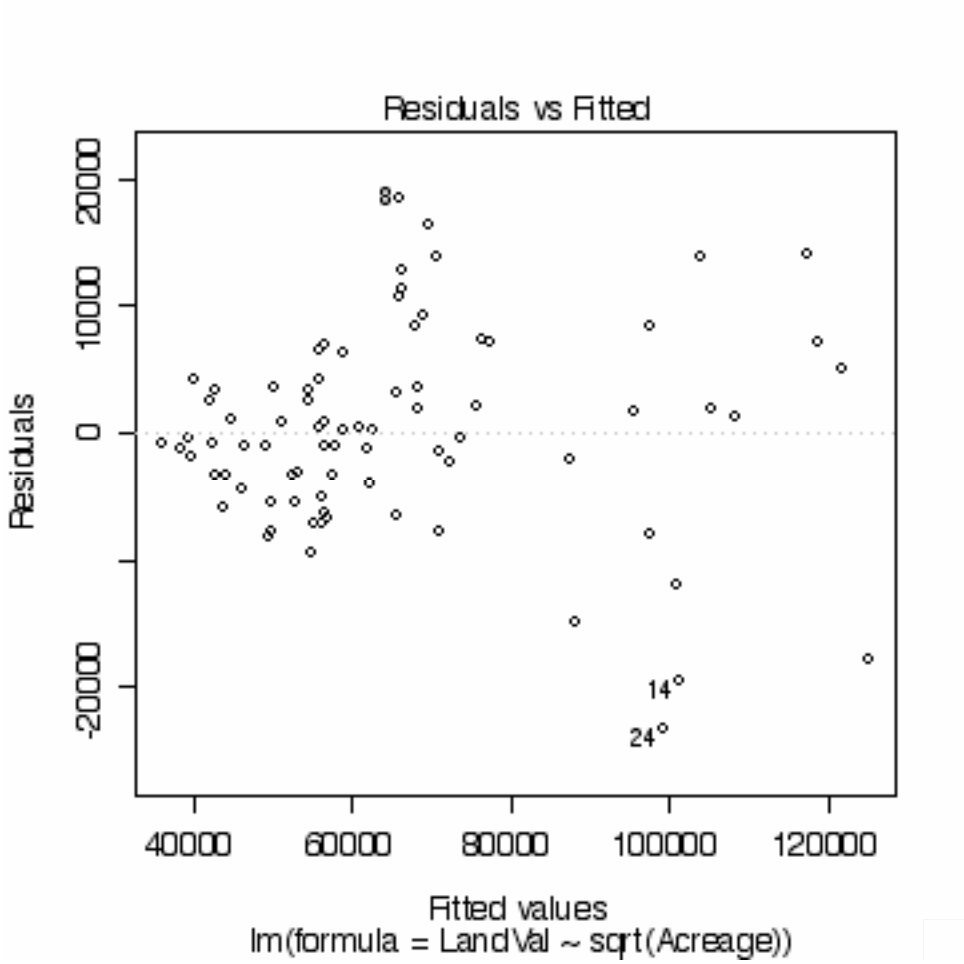
```
boxcox(LandVal ~ Acreage)
```



This plot tells us that L is some number between about 1/2 and 1.5. Note that if L = 0 then we would take the log transformation. So what I get out of this is that I'm right to ignore log transforms, and my idea of a square root (x^.5) is not ridiculous.

So lets fit that model:

```
> fit.sqrt <- lm(LandVal ~sqrt(Acreage))
> plot(fit.sqrt)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```



Ahh. Trouble. Notice the "fan" shape. Residuals near fitted-value = 40K are close to the line, and they get more and more spread out for larger fitted values. This is bad and means we'll have incorrect confidence intervals and p-values. (Why? The reason is that the precision of any estimate depends on the variation in the observation. If the observations are all very close together, then we should be able to come up with an estimate of the mean that's very close to the true value. For these data, we should be able to get precise estimates for lower-valued properties because there is less variability, where for higher valued properties our estimates will not be as precise because of the increased variability. However, our regression model assumes that variability is the same, and therefore confidence intervals will be too wide at the low-valued properties and too small at the high.) The good news

is that the qqnorm plot shows that the residuals are approximately normally distributed.

This is probably the best we can hope for here.(I tried other transformations. For example, a fit of sqrt(LandVal)~sqrt(Acreage), but this results in a non-linear fit. Usually, it's better to violate this assumption that the residuals have a constant scatter than it is to violate the assumption of linearity (although it might depend on your purpose.) One reason for why this is better is that there are other methods (called "weighted least squares" that can deal with data in which the residuals' standard error increases, as it does here.)

It shouldn't be surprising, by the way, that the residuals have greater standard error at high values than low. It makes sense that there would be more variability among expensive properties than among inexpensive properties.

Our "best" model (with all of its flaws) is
LandVal = 10826 + 47059*sqrt(Acreage)

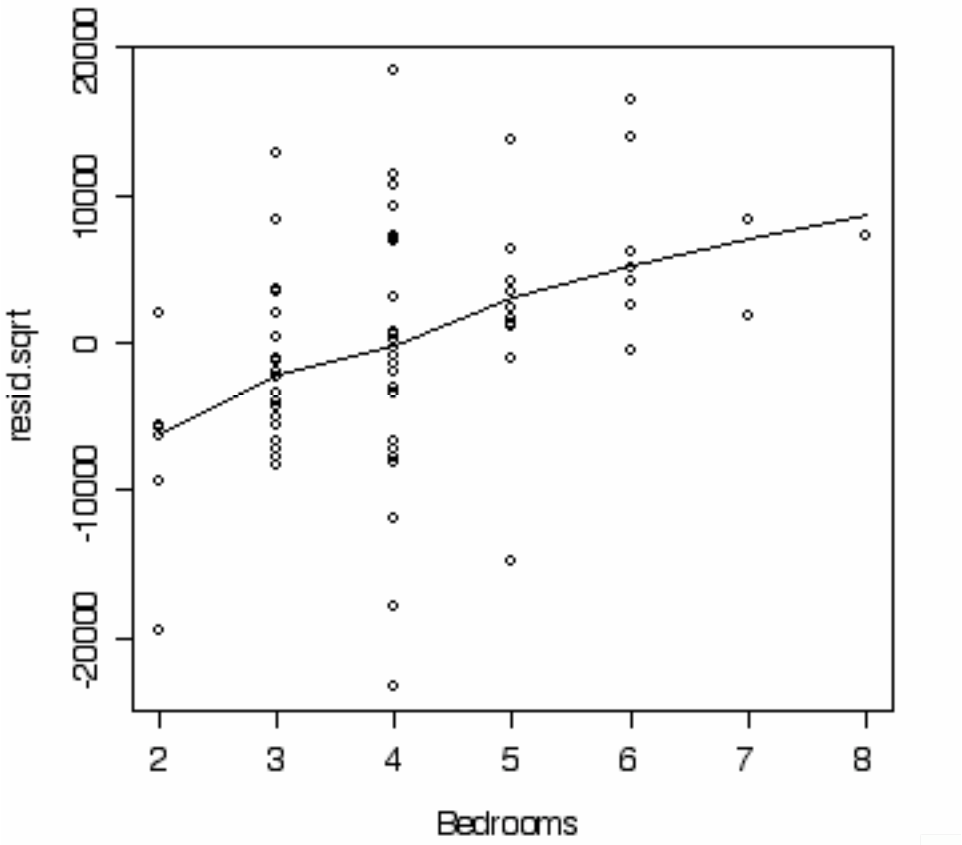
From this we learn that the more valuable lots do tend to be those with more area. The fact that this relationship is non-linear with respect to the amount of acreage makes it hard to give a simple "for every one-unit difference in acreage" sort of answer.

iii) Choose any other variable (say "height") that you think might be related to the value of land. Plot the residuals from (ii) against this variable. Describe what you see. Do you think that this variable would be useful additional information for predicting the value of land?

It seems reasonable (at least to me) that the value would depend a lot on the number of bedrooms in a house. So I chose to plot the residuals against the Bedrooms variable.

Note: had you chosen "Height" (which turns out to be boring), you would conclude that there was no relation between Height and Land Value (once we had already taken lot-size into account) because you would see no pattern in the residuals.

```
> resid.sqrt <- residuals(fit.sqrt)
> plot(Bedrooms, resid.sqrt)
> lines(lowess(Bedrooms, resid.sqrt))
```



This plot shows that there is a trend . The residuals are bigger for properties wiht more bedrooms. This suggests that the number of bedrooms does in part determine the value of the land. More particularly, it tells us that for our model, we were underestimating the value for homes with many bedrooms, and overestimating for rooms with a smaller number of bedrooms.

iv) The slope in (ii) relates the mean land value to the size of the lot. What assumptions are needed in order to make inferences on this slope? Do you think they are valid here? Find a 95% confidence interval for the slope and interpret.

We assume that the residuals are normally distributed (which is approximately true) and that the standard error of the residuals is

the same across all lots (which is not true), and that the errors are independent of each other (which we don't really know enough to decide.) We haven't discussed this, but the result

The 95% CI is of the form
estimate +/- t * SE

We get the value of the estimate (47059) and its standard error (1914) from the summary printout (summary(fit.sqrt)).

We choose t so that, in a t-distribution with n-2 degrees of freedom, .05/2 = .025 area will be above it. Or, put differently, .975 will be below it. Here's how:

```
> n <- length(LandVal)
> n
[1] 79
> qt(.975,77)
[1] 1.991254
```

Now we can calculate the confidence interval for the slope:

```
> margin.of.error <- qt(.975,77)*1914
> lb <- 47059 - margin.of.error
> ub <- 47059+margin.of.error
> lb
[1] 43247.74
> ub
[1] 50870.26
```

So our 95% confidence interval for the true slope is ($42,247.74, $50870.26). Which means that we are confident the slope -- which relates the square-root of the size of the lot with the value of the lot -- is in this range.

v) Estimate the mean value of properties that sit on .80 acres. Find a 95% confidence interval. Do you think the assumptions required to make this inference are satisfied? Explain.

To do this we need to create a dataframe with the size of the lot for which we want to estimate value. This is a little tricky, since our

model fits a square-root. R makes this pretty straight-forward though.

```
> new.lotsize <- data.frame(Acreage=c(.8))
> predict.lm(fit.sqrt,new.lotsize, interval="confidence")
          fit     lwr      upr
[1,] 52917.14 50882.5 54951.79
```

We tell R to set the interval to "confidence" because we are estimating a mean. Our interpretation is that, of all lots on land with .8 acres, we are 95% confident the mean value is between $50882 and $54951.

The assumptions are not satisfied, because of the non-constant variance.

You can check that this works "by hand". The "fit" term is found just from plugging into the regression equation:

LandVal = 10826 + 47059*sqrt(Acreage)
= 10826 + 47059*sqrt(.8) = 52917

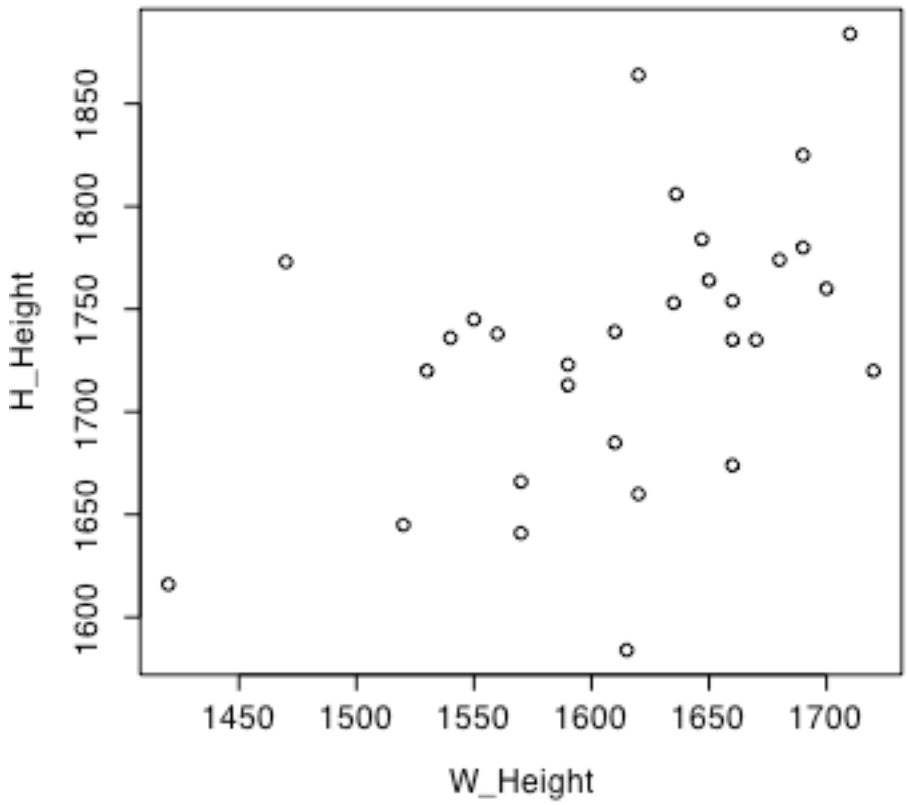
vi) Predict the value of a property that sits on 1.7 acres of land. Find a 95% confidence interval. Again, discuss whether the data can support this inference. (That is, are the assumptions valid?)

This time we're asked to predict the value of a single lot of land, and so we make a prediction interval:

```
> new.lotsize <- data.frame(Acreage=c(1.7))
> predict.lm(fit.sqrt,new.lotsize, interval="prediction")
          fit      lwr      upr
[1,] 72183.86 56594.55 87773.17
```

We're 95% confident that a single property with 1.7 acres will be valued between $56594 and $87773).

C. Do opposites attract? A random sample of married couples in Great Britain might help answer this question (at least with respect to a few variables.)
i) Use a regression to predict husband's height given his wife's height. Is there a relationship?



The first step in any regression is to make a plot. It appears there is a positive, linear association, which means that taller women tend to marry taller men (at least in Great Britian). But this is a random sample of couples, and so it is possible that what appears to be a "real" association is in fact just due to the variability inherent in random sampling. And so we examine a regression line.

```
> wife.fit <- lm(H_Height~W_Height)
> summary(wife.fit)

Call:
lm(formula = H_Height ~ W_Height)

Residuals:
    Min      1Q  Median      3Q     Max
-150.03  -43.25    3.69   33.18  127.77

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1025.3659   256.1259   4.003 0.000416 ***
W_Height       0.4388     0.1586   2.766 0.009929 **
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 61.11 on 28 degrees of freedom
```

Multiple R-Squared: 0.2146, Adjusted R-squared: 0.1866
F-statistic: 7.652 on 1 and 28 DF, p-value: 0.009929

Assuming that the necessary assumptions hold (which we discuss in the next part), we see that the p-value associated with the slope (which is given in the table in the column labelled "Pr(>|t|)") that because the p-value is less than .05, we reject the null hypothesis that the true slope is 0 and conclude that, in fact, the true slope is not zero. So there does seem to be a relationship between heights.

ii) Interpret the slope and find a 95% confidence interval. Interpret the confidence interval, and discuss whether the assumptions that are required to make this interval interpretable are true. (Heights are measured in milimeters).

This means that women who are 1 mm taller tend to have husbands who are .4388 mm taller, on average.

A 95% CI is given by the formula estimate +/- t * SE

t comes from a t-distribution with n-2 degrees of freedom, and is determined so that it has 2.5% of the area in the distribution above it or, equivalently, 97.5% below it:

```
> n <- length(W_Height)
> n
[1] 30
> qt(.975,28)
[1] 2.048407
```

The standard error we get from the summary print-out: 0.1586.

So the commands below calculate the margin of error and the upper and lower limits of the confidence interval:

```
> me <- qt(.975,28)*.1586
> me
[1] 0.3248774
> .4388-me
[1] 0.1139226
> .4388+me
[1] 0.7636774
>
```

So we are 95% confident that the true value of the slope (the value if we were to examine all married couples in Great Britain) is between (.114 and .764).

This inference requires that the linear model hold, that the errors be independent and normally distributed, and that the standard deviation of the errors be constant. The command plot(wife.fit) steps us through the diagnostic plots we need to assess this. The plots suggest that the assumptions are fairly sound.

iii) Identify any influential points. What happens if these are removed?

The final plot in the plot(wife.fit) command shows the Cook's distances, which measure influence. These suggest that couple 25 and couple 20 are particularly influential:

```
> cbind(W_Height, H_Height)[c(25,20),]
     W_Height H_Height
[1,]     1470     1773
[2,]     1710     1884
```

iv) According to the model, among all wives who are 1680 mm tall, about 68% of the husbands are between ? and ? in height?

Our least squares regression tells us that we expect the husband to be 1025.4 + 1680*.4388 = 1762.4, give or take sigma --the standard deviation of the errors. This value is given in the summary printout as 61.11. So we would expect 68% of the people to be within one standard deviation of the expected value or 1762.4 +/- 61.1 or 1701.3 to 1823.5

iv) Play matchmaker: predict (with a confidence interval) how tall the husband of a woman who is 1650 mm tall should be.

Before using the predict.lm command, we need to create a new dataframe that has the value of 1650. Next, we use the predict.lm command, telling it to create a prediction interval (since we are finding a confidence interval for an individual, and not for a mean):

```
> wife <- data.frame(W_Height=c(1650))
> predict.lm(wife.fit,wife,interval="prediction")
         fit      lwr      upr
[1,] 1749.392 1621.583 1877.201
```

The 95% prediction interval is (1621.6, 1877.2). We are 95% confident the husband's height will be within these limits.

v) A psychologist claims that short men feel the need to compensate by marrying taller wives. As evidence, he looks only at the "shorter" men in the sample: those less than the first quartile in height (which is those less than 1692). He finds that on average, these men are 1.28 standard deviations shorter than average. However, their wives are only .47 standard deviations shorter than average. This, he claims, is evidence that short men compensate by looking for taller wives. (Not wives who are taller than them, mind you, but wives who are taller with respect to other wives.) Critique this reasoning.

This is an example of the regression to the mean phenomenon. Men who are shorter than average will tend to have wives who are closer to average then the men. The psychologist isn't considering the other side of the coin: men who are taller than average will tend to have wives who are closer to average height then their husbands. It also works the other way. Women who are shorter than average will tend to have husband's who are closer to average.