Solutions to Inferences for Linear Regression Homework

1. (a)

```
> data <- read.table("CH01PR19.txt")
> head(data)
     V1 V2
1 3.897 21
2 3.885 14
3 3.778 28
4 2.540 22
5 3.028 21
6 3.865 31
> names(data) <- c("GPA", "ACT")
> head(data)
    GPA ACT
1 3.885  14
2 3.778  28
3 2.540  22
4 3.028  21
5 3.865  31
6 2.962  32
> out <- lm(data$GPA~data$ACT)
> summary(out)

Call:
lm(formula = data$GPA ~ data$ACT)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.73842 -0.32556  0.04421  0.44644  1.25203 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.06789    0.32025   6.457 2.53e-09 ***
data$ACT     0.04036    0.01273   3.170  0.00194 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6193 on 117 degrees of freedom
Multiple R-squared:  0.07911,	Adjusted R-squared:  0.07124 
F-statistic: 10.05 on 1 and 117 DF,  p-value: 0.001944

> confint(out, level=.99)
                  0.5 %     99.5 %
(Intercept) 1.229306076 2.90646976
data$ACT    0.007026265 0.07370037
```

We are about 99% confident that the actual intercept parameter beta_0 is between 1.229 and 2.9065. We can be about 99% confident that the actual slope parameter beta_1 is between 0.0070 and 0.0737. This interval does *not* include 0. The significance of this is that the slope of the true linear model is probably different from 0, indicating there is likely some kind of relationship between the response (first-year GPA) and the predictor (ACT score).

1. (b) I actually did this above with the lm() command. Remember from the notes that testing to see if there exists a linear association or not is equivalent to testing whether or not the slope coefficient beta_1 is 0 (since b_1 = r x s_y / s_x). The p-value for this test is 0.00194 (see the output above in part (a), indicating the beta probably really does differ from 0.

1. (c) So I'll do the same thing I did in part (a) but change the confidence level to 98%:

```
> confint(out, level=.98)
                  1 %       99 %
(Intercept) 1.31252739 2.82324845
data$ACT    0.01033465 0.07039199
```

   We can be about 98% confident that the actual intercept beta_0 is between 1.3125 and 2.8233. The intercept tells us the predicted value of first-year GPA for an ACT score of 0. This is sort of silly, right? Who gets a 0 on the ACT? Also, if someone did score that low, would they be likely to get a first-year GPA between 1.3 and 2.9? Maybe the linear model is no good for predicting GPA for really low ACT scores.

1. (d)

```
> attach(data)
> out <- lm(data)
> nd = data.frame(ACT=29)
> predict(out, nd, interval="confidence", level=.95)
       fit     lwr      upr
1 3.238424 3.08322 3.393629
```

We can be about 95% confident that the actual mean first-year GPA for students matriculating with ACT scores of 29 is between 3.08322 and 3.393629.

1. (e)

```
> predict(out, nd, interval="prediction", level=.95)
       fit      lwr      upr
1 3.238424 2.002225 4.474624
```

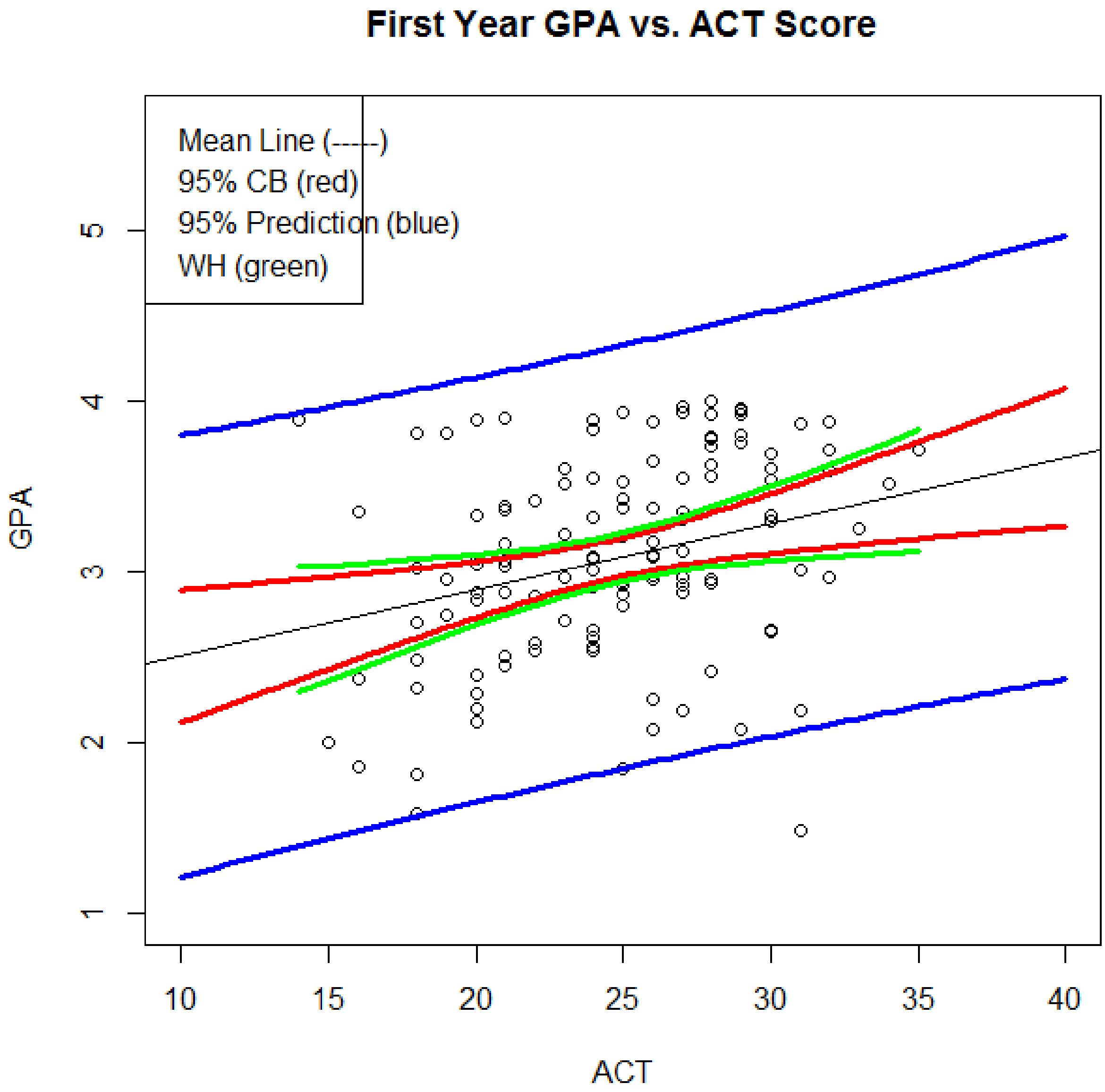
We can be about 95% confident that Billy's first-year GPA will fall somewhere between 2.002225 and 4.474624. Note that technically our normality assumption must be violated here because normal curves never come down to the horizontal axis, and yet GPA is capped, both from above and below. Perhaps this means we should use some other kind of model… more about checking model assumptions next week!

1. (f) This prediction interval is wider, and it should be. The prediction interval comes straight from the conditional distribution of the response given the predictor variable value. On the other hand, we know that sample means have a smaller variance than that of the original

distribution… in fact they differ by a factor of 1/n, and so the CIs for means should always be smaller than the prediction intervals.

1. (g) and (h) So in parts (g) and (h)… these may be a little unclear. I'll just do them together, and make three plots on one graph- (1) confidence interval bands for mean responses; (2) prediction interval bands; (3) Working-Hotelling bands:

```
x <- data$ACT
y <- data$GPA
d <- data.frame(x, y)
out <- lm(y~x, data=d)
nd <- data.frame(x=seq(10, 40, length=119)) ### need length same as
################################################# number of rows in
################################################# conf and pred
conf <- predict(out, interval = "confidence", newdata=nd, level = .95)
pred <- predict(out, interval = "prediction", newdata=nd, level = .95)
plot(y~x, data = d, ylim=c(1, 5.6), xlim=c(10, 40), main="First Year GPA vs. ACT Score",
xlab="ACT", ylab="GPA")
abline(out)
lines(nd$x, conf[,2], col="red", lwd=3)
lines(nd$x, conf[,3], col="red", lwd=3)
lines(nd$x, pred[,2], col="blue", lwd=3)
lines(nd$x, pred[,3], col="blue", lwd=3)
## What follows is for the Working-Hotelling bands for the entire regression line
CI <- predict(out, se.fit=TRUE)
W <- sqrt(2*qf(0.95,length(out$coefficients), out$df.residual))
Band <- cbind(CI$fit - W * CI$se.fit, CI$fit + W * CI$se.fit )
points(sort(data$ACT), sort(Band[,1]), type="l", lty=1, col="green", lwd=3)
points(sort(data$ACT), sort(Band[,2]), type="l", lty=1, col="green", lwd=3)
legend("topleft", legend=c("Mean Line (-----)", "95% CB (red)",
"95% Prediction (blue)", "WH (green)"), col=c(1,1))
```


First Year GPA vs. ACT Score
Mean Line (-----)
95% CB (red)
95% Prediction (blue)
WH (green)
GPA
1
2
3
4
5
ACT
10
15
20
25
30
35
40

2. The F-test is only good for performing a two-tailed test for beta_1. You can only run left- or right-tailed tests for beta_1 when doing a t-test. Also, the F-test is one-tailed because when testing H_a: beta_1 different from 0, this is a two-tailed t-test, and the p-value will be the sum of the tail areas (on both sides) from a t-distribution with n-2 degrees of freedom. But recall that an F-random variable with 1 numerator degree of freedom and n-2 denominator degrees of freedom behaves exactly like the square of a t-random variable with n-2 denominator degrees of freedom. So the tail are falls completely to the right under the F-distribution.

3. The Bonferroni inequality says that for any events $E_1, E_2, \ldots, E_n$, that

$$P\left(\bigcap_{i=1}^{n} E_i\right) \geq \sum_{i=1}^{n} P(E_i) - n + 1 .$$

Proof by Mathematical Induction.

Pf: $P(E_1) = \sum_{i=1}^{1} P(E_i) - \underline{1} + 1$,

so the proposition holds when $i = 1$. Now assume it is true for $i \in \{1, 2, \ldots, k\}$. Then

$$P\left(\bigcap_{i=1}^{k+1} E_i\right) = P\left(\left(\bigcap_{i=1}^{k} E_i\right) \cap E_{k+1}\right)$$

$P(A\cap B) = P(A) + P(B) - P(A \cup B)$

$$= \underbrace{P\left(\bigcap_{i=1}^{k} E_i\right)}_{\geq \sum_{i=1}^{k} P(E_i) - k + 1} + P(E_{k+1}) - P\left(\left(\bigcap_{i=1}^{k} E_i\right) \cup E_{k+1}\right)$$

by assumption

$$\geq \sum_{i=1}^{k} P(E_i) - k + 1 + P(E_{k+1}) - P\left(\left(\bigcap_{i=1}^{k} E_i\right) \cup E_{k+1}\right)$$

$$= \sum_{i=1}^{k+1} P(E_i) - k + 1 - \underbrace{P\left(\left(\bigcap_{i=1}^{k} E_i\right) \cup E_{k+1}\right)}_{\leq 1} \geq \sum_{i=1}^{k+1} P(E_i) - (k+1) + 1$$

4. Since we seek two CIs, and we want the family error rate to be no more than 5% (so that the family confidence level is 95%), we should make the individual CIs at the 97.5% confidence level each. You can see this if you use the Bonferroni inequality, and use .95 as P(E1∩E2), then solve the inequality for the sum (P(E_i)) and divide by 2 (since we want each individual probability to be the same):

```
> data <- read.table("CH01TA01.txt")
> head(data)
  V1  V2
1 80 399
2 30 121
3 50 221
4 90 376
5 70 361
6 60 224
> names(data) <- c("LotSize", "WorkHours")
> head(data)
  LotSize WorkHours
1      80       399
2      30       121
3      50       221
4      90       376
5      70       361
6      60       224
> out<- lm(data[,2]~data[,1])
> out

Call:
lm(formula = data[, 2] ~ data[, 1])

Coefficients:
(Intercept)    data[, 1]
      62.37         3.57
```

So the equation of the regression line is

hat(WorkHours) = LotSize * 3.57 + 62.37

We can get the CIs by doing

```
> confint(out, level=.975)
                1.25 %    98.75 %
(Intercept) -0.4043574 125.136075
data[, 1]    2.7382061   4.402198
```

That is, we can be about (or maybe at least??? Remember the Bonferroni procedure gives us a theoretical lower bound on the family error rate) 95% confident that the slope is between 2.738 and 4.4022 AND that the intercept is between -.404357 and 125.136.

Also, just a quick note about CIs in general… from now on, when I make them, I will try to remember to round the lower endpoints down and upper endpoints up… why would you want to round CI endpoints this way?

5. (a) So the null and alternative hypotheses should be

H_0: beta_1 = 3.0
H_a: beta_1 IS NOT EQUAL TO 3.0

Here's my R code. Note how I've multiplied a left-tail area by 2 to get the p-value for a two-tailed test:

```
> res <- out$residual
> SSE <- sum(res^2)
> MSE <- SSE / (25-2)
> MSE
[1] 2383.716
> sqrt(MSE)
[1] 48.82331
> stdev.b1 <- sqrt(MSE/sum((data$LotSize - mean(data$LotSize))^2))
> test.statistic <- (3.570-3)/stdev.b1
> test.statistic
[1] 1.642783
> p-value <- 2*pt(-test.statistic, 25-2)
Error in p - value <- 2 * pt(-test.statistic, 25 - 2) :
  object 'p' not found
> p-value <- 2*pt(-test.statistic, 23)
Error in p - value <- 2 * pt(-test.statistic, 23) : object 'p' not found
> p-value <- 2*pt(-1*test.statistic, 23)
Error in p - value <- 2 * pt(-1 * test.statistic, 23) :
  object 'p' not found
> p-value <- 2*pt(-1.642783, 23)
Error in p - value <- 2 * pt(-1.642783, 23) : object 'p' not found
> p.value <- 2*pt(-1.642783, 23)
> p.value
[1] 0.1140309
```

This p-value is not that small… remember the meaning of a p-value: THE P-VALUE IS THE CHANCE YOU'D SEE SOMETHING AS STRANGE AS YOU SAW IF INDEED THE NULL HYPOTHESIS IS ACTUALLY TRUE. So what this means is that if indeed the actual beta-1 value equals 3, then there is about an 11.4% chance we would see a test statistic as strange as the one we saw when taking a random sample of responses for the x-values (lot sizes in this case) we did. 11% chance is not that "strange". I would fail to reject the null in this case: there is insufficient evidence to conclude that beta_1 is not equal to 3. The text says to use alpha=5%. Of course, since the p-value > 5%, definitely fail to reject the null.

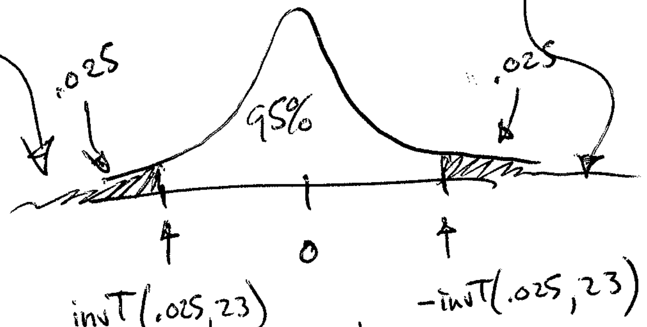
#5. (b) So if β_1 actually equals 3.5, we want to compute the power of the test, which is the chance of rejecting $H_0: \beta_1 = 3.0$ when β_1 indeed equals 3.5. Sorry for the redundancy!

b_1 is random

$$\text{Power} = P\left(\frac{b_1-3.0}{.35} \in R.R. \,\middle|\, \frac{b_1-3.5}{.35} \sim t \text{ dist. w/ 23 d.f.}\right)$$

We're assuming $\sigma(b_1) = .35$

Rejection Region ... $\left(\text{invT}(.025, 23), \text{invT}(.975, 23)\right)^c$



$$= P\left(\frac{b_1-3}{.35} \le \text{invT}(.025,23) \ \underline{\text{OR}}\ \frac{b_1-3}{.35} \ge \text{invT}(.975,23) \,\middle|\, \frac{b_1-3.5}{.35} \sim t_{df:23}\right)$$

$$= P\left(\frac{b_1-3.5}{.35} \le \text{invT}(.025,23) - \frac{.5}{.35} \ \underline{\text{OR}}\ \frac{b_1-3.5}{.35} \ge \text{invT}(.975,23) - \frac{.5}{.35} \,\middle|\, \frac{b_1-3.5}{.35} \sim t_{df=23}\right)$$

$$= \text{tcdf}\left(-10^{99}, \text{invT}(.025,23) - \frac{1}{.7}, 23\right) + \text{tcdf}\left(\text{invT}(.975,23) - \frac{1}{.7}, 10^{99}, 23\right)$$

$$= .0186 + .03329 = \boxed{.05189}$$

I'm using my TI-84 plus for this one!

So if $\beta_1 = 3.5$, the power is pretty low... only @ about 5%. If β_1 is even higher, then we should stand a better chance of detecting... that is, the power of the test should increase as the actual value of β_1 gets farther from its proposed value under H_0.

For $\beta_1 = 3.0 + 1 = 4$ ...

$$\text{Power} = P\left(\frac{b_1 - 4}{.35} \le \text{invT}(.025, 23) - \frac{2}{7} \ \underline{\text{OR}} \ \frac{b_1 - 4}{.35} \ge \text{invT}(.975, 23) - \frac{2}{7} \ \middle| \ \frac{b_1 - 4}{.35} \sim t_{df=23}\right)$$

$$= \text{tcdf}\left(-10^{99}, \text{invT}(.025, 23) - \frac{2}{7}, 23\right) + \text{tcdf}\left(\text{invT}(.975, 23) - \frac{2}{7}, 10^{99}, 23\right)$$

$$= .0576$$

For $\beta_1 = 4.5$ ...

$$\text{Power} = \text{tcdf}\left(-10^{99}, \text{invT}(.025, 23) - \frac{3}{7}, 23\right) + \text{tcdf}\left(\text{invT}(.975, 23) - \frac{2}{7}, 10^{99}, 23\right)$$

$$\approx 6.7\%$$

5(c) The F* test statistic is an F statistic, and only applies to two-tailed tests about beta_1 = 0.

6. For a family error rate of 10%, the family confidence level is 90%. Using the Bonferroni inequality, the individual confidence levels will all be (since we're to make three of them)

   (.90 +3 – 1)/3 = .9666666666666666666666…

   and so I will use .96 (round down) for the individual rates.

```
> attach(data)
> out<- lm(WorkHours ~ LotSize)
> nd <- data.frame(LotSize = c(30,60,100))
> predict(out, nd, interval="confidence", level = .966)
      fit      lwr      upr
1 169.4719 131.2158 207.7281
2 276.5780 253.2164 299.9396
3 419.3861 387.2109 451.5612
```

Here's my statement: "We can be about 90% confident that the mean work hours for lot sizes of 30, 60, and 100 are within these respective intervals:

[131.21579, 207.7281], [253.21636, 299,9396], and [387.21090, 451.5612]."

Here's my code for the Working-Hotelling:

```
> CI <- predict(out, se.fit=TRUE)
> W <- sqrt(2*qf(.90, length(out$coefficients), out$df.residual))
> Band <- cbind(CI$fit - W*CI$se.fit, CI$fit + W*CI$se.fit)
> cbind(Band, LotSize)
                              LotSize
1  324.58279 371.3813     80
2  131.15419 207.7897     30
3  213.82658 267.9253     50
4  356.63466 410.7334     90
5  290.23136 334.3286     70
6  253.17875 299.9772     60
7  445.83809 535.7421    120
8  324.58279 371.3813     80
9  387.15910 451.6130    100
10 213.82658 267.9253     50
11 172.94698 237.4009     40
12 290.23136 334.3286     70
13 356.63466 410.7334     90
14  88.81789 178.7219     20
15 416.77035 493.4058    110
16 387.15910 451.6130    100
17 131.15419 207.7897     30
18 213.82658 267.9253     50
19 356.63466 410.7334     90
```

```
20 416.77035 493.4058     110
21 131.15419 207.7897      30
22 356.63466 410.7334      90
23 172.94698 237.4009      40
24 324.58279 371.3813      80
25 290.23136 334.3286      70
```

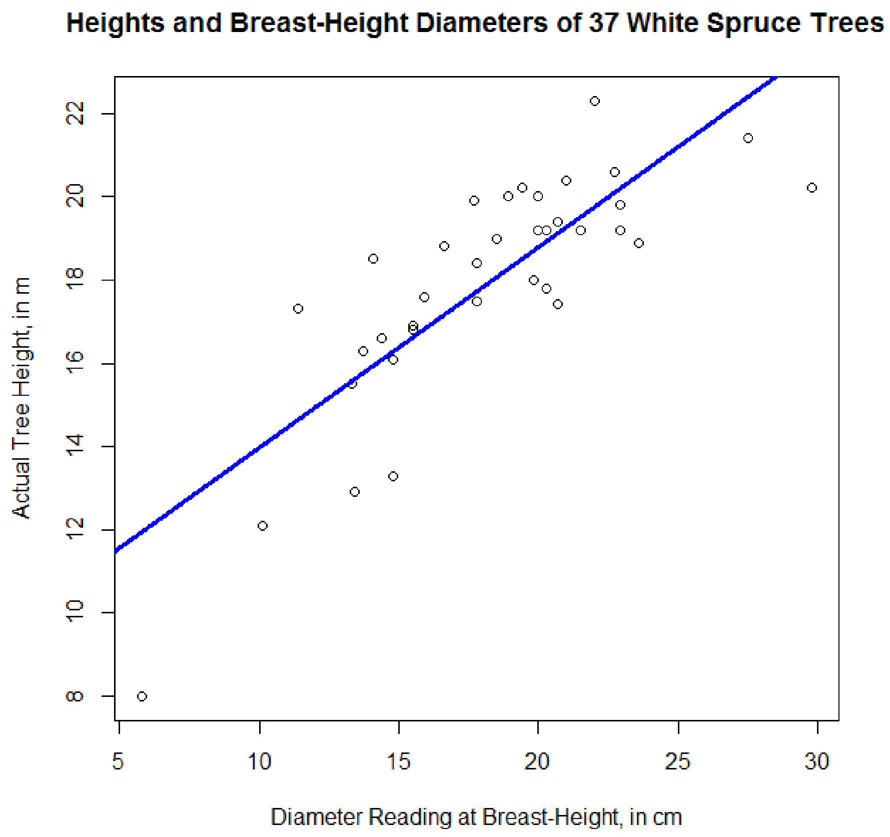
So you see the Working-Hotelling intervals are (for the 30, 60, 100 lot sizes, respectively)

[131.15419, 207.7897], [253.17875, 299.9772], and [387.15910, 451.6130].

7.

```
> attach(ws)
> out <- lm(ht ~ diam)
> plot(diam, ht, main = "Heights and Breast-Height Diameters of 37 White Spruce Trees",
+ xlab = "Diameter Reading at Breast-Height, in cm",
+ ylab = "Actual Tree Height, in m")
> abline(out, col = "blue", lwd = 3)
> anova(out)
Analysis of Variance Table

Response: ht
          Df  Sum Sq Mean Sq F value    Pr(>F)
diam       1 183.245 183.245  65.101 2.089e-09 ***
Residuals 34  95.703   2.815
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



The ANOVA F-test for H_a: beta_1 not equal to 0 gives a p-value of 2.089*10^-9, indicating that beta_1 is probably not equal to 0. Note you can also view the t-test(s):

```
> summary(out)
```

```
Call:
lm(formula = ht ~ diam)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.9394 -0.9763  0.2829  0.9950  2.6644 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  9.14684    1.12131   8.157 1.63e-09 ***
diam         0.48147    0.05967   8.069 2.09e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.678 on 34 degrees of freedom
Multiple R-squared:  0.6569,    Adjusted R-squared:  0.6468 
F-statistic:  65.1 on 1 and 34 DF,  p-value: 2.089e-09
```

The ANOVA F-test output is on the very last line here, and is the same result as before. The p-value for the t-test for beta_1 is 2.09e-09… the same! That's because, remember, that when you square a t-random variable with d degrees of freedom you get an F random variable with 1 numerator and d denominator degrees of freedom. The t-test statistic there is 8.069. If you square it, you get the value of the F-statistic.

The null and alternative hypotheses around the beta_1 parameter in this case are, of course

H_0: beta_1 = 0
H_a: beta_1 differs from 0

Actually, from looking at r^2 and the plot, it seems there is a decent linear relationship between the two variables.