Assign. 1 STA 445

April Meadows

2024-02-22

```
library(tidyverse)
library(ggplot2)
```

Directions:

This assignment covers chapter 5. Please show all work in this document and knit your final draft into a pdf. This is assignment is about statistical models, which will be helpful if you plan on taking STA 570, STA 371, or STA 571.

Problem 1: Two Sample t-test

a. Load the `iris` dataset.

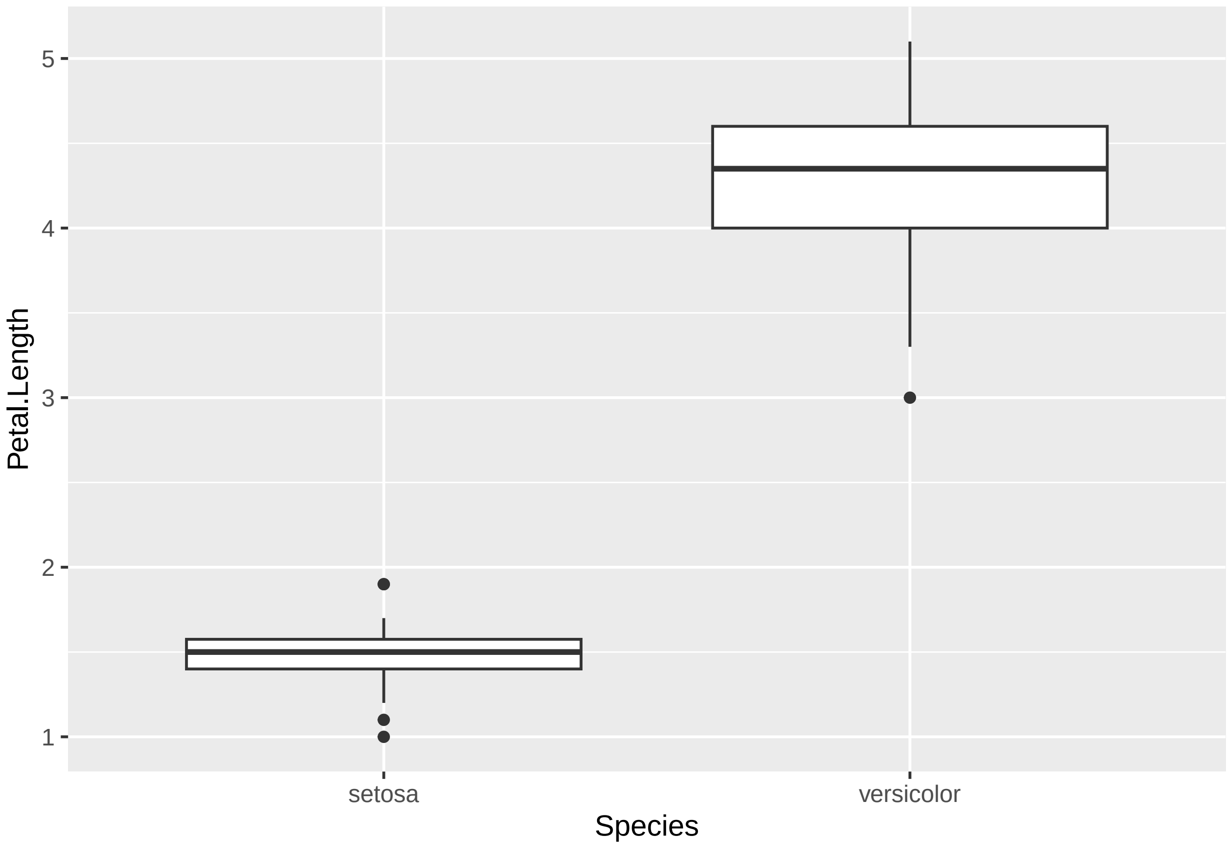
```
# loads in the iris data
data("iris")
```

b. Create a subset of the data that just contains rows for the two species setosa and versicolor using filter. Use slice_sample to print out 20 random rows of the dataset.

```
# creates a subset of iris data with only setosa and versicolor species
## omits virginica species to do this, **find a way to do it better**
subiris <- iris %>%
  filter(Species!="virginica")
```

c. Create a box plot of the petal lengths for these two species using ggplot. Does it look like the mean petal length varies by species?

```
# creates a boxplot of petal length for each species
subiris %>%
  ggplot(aes(x=Species,y=Petal.Length)) +
    geom_boxplot()
```



It does look like the mean petal length for each species is different.

d. Do a two sample t-test using t.test to determine formally if the petal lengths differ. Note: The book uses the tidy function in the broom package to make the output "nice". I hate it! Please don't use tidy.

```
# performs a two sample t-test on the subiris data with a confidence level of 90%
t.test(Petal.Length ~ Species, data=subiris, conf.level=0.9)
```

```
##
##  Welch Two Sample t-test
##
## data:  Petal.Length by Species
## t = -39.493, df = 62.14, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group setosa and group versicolor is not equ
## 90 percent confidence interval:
##  -2.916299 -2.679701
## sample estimates:
##    mean in group setosa mean in group versicolor
##                   1.462                    4.260
```

d. What is the p-value for the test? What do you conclude?

The p-value for this test is very close to zero, which is less than 0.05. We reject the null hypothesis. There is sufficient evidence to suggest that the petal length means between the two species are different. The 90% confidence interval is (-2.916,-2.680), so we can say with 90% confidence that the difference in the means lies within that interval.

e. Give a 95% confidence interval for the difference in the mean petal lengths.

```r
# performs a two sample t-test on the subiris data with a confidence level of 95% (the default)
t.test(Petal.Length ~ Species, data=subiris)
```

```
##
##  Welch Two Sample t-test
##
## data:  Petal.Length by Species
## t = -39.493, df = 62.14, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group setosa and group versicolor is not equ
## 95 percent confidence interval:
##  -2.939618 -2.656382
## sample estimates:
##     mean in group setosa mean in group versicolor
##                    1.462                    4.260
```

Once again, the p-value for this test is very close to zero, which is less than 0.05. We reject the null hypothesis. There is sufficient evidence to suggest that the petal length means between the two species are different. The 95% confidence interval is (-2.940,-2.656), so we can say with 95% confidence that the difference in the means lies within that interval.

f. Give a 99% confidence interval for the difference in mean petal lengths. (Hint: type ?t.test. See that you can change the confidence level using the option conf.level)

```r
# performs a two sample t-test on the subiris data with a confidence level of 99%
t.test(Petal.Length ~ Species, data=subiris, conf.level=0.99)
```

```
##
##  Welch Two Sample t-test
##
## data:  Petal.Length by Species
## t = -39.493, df = 62.14, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group setosa and group versicolor is not equ
## 99 percent confidence interval:
##  -2.986265 -2.609735
## sample estimates:
##     mean in group setosa mean in group versicolor
##                    1.462                    4.260
```

Once again, the p-value for this test is very close to zero, which is less than 0.05. We reject the null hypothesis. There is sufficient evidence to suggest that the petal length means between the two species are different. The 99% confidence interval is (-2.986,-2.610), so we can say with 99% confidence that the difference in the means lies within that interval.

g. What is the mean petal length for setosa?

```r
# finds the mean petal length for the setosa species in the subiris data
subiris %>%
  filter(Species=="setosa") %>%
  summarise(setosa_mean=mean(Petal.Length))
```

```
##   setosa_mean
## 1       1.462
```

h. What is the mean petal length for versicolor?

```r
# finds the mean petal length for the versicolor species in the subiris data
subiris %>%
  filter(Species=="versicolor") %>%
```

```
  summarise(versicolor_mean=mean(Petal.Length))
```

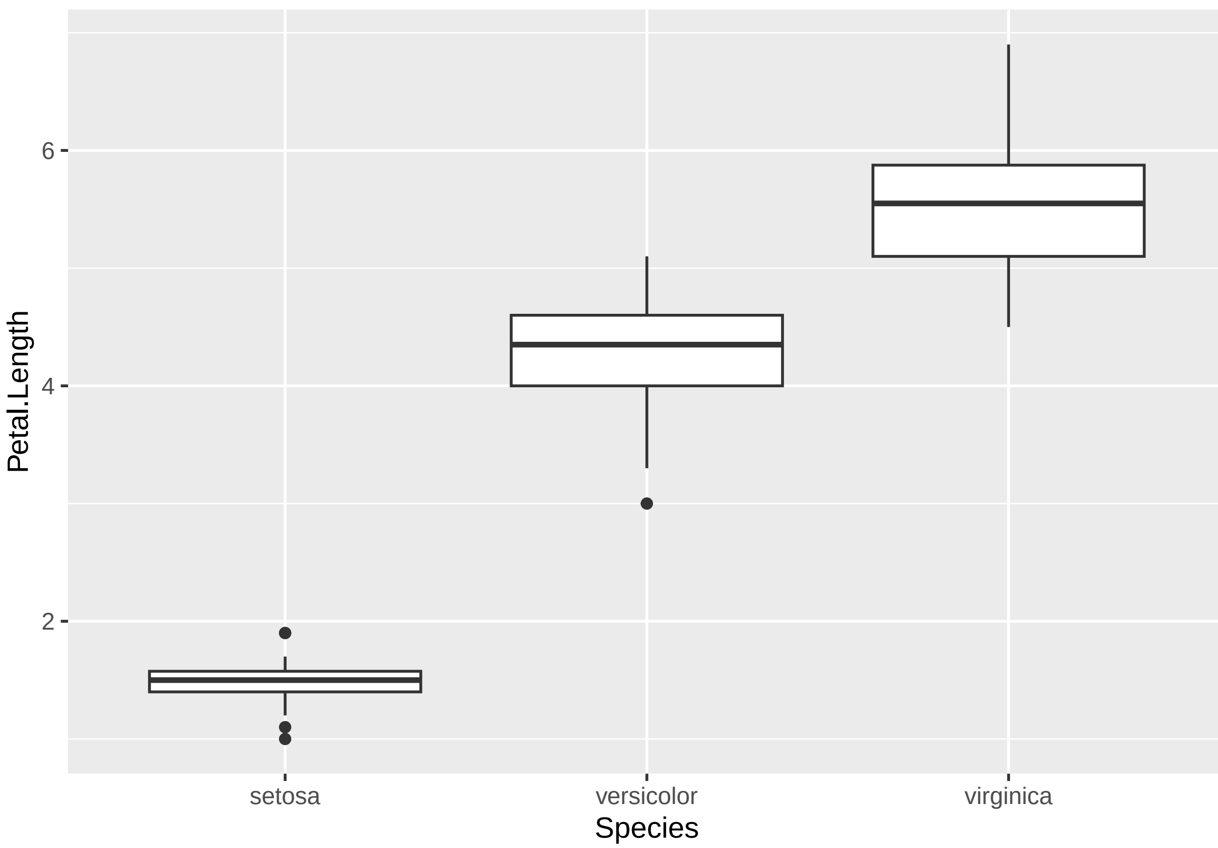
```
##   versicolor_mean
## 1            4.26
```

Problem 2: ANOVA

Use the iris data with all three species.

a. Create a box plot of the petal lengths for all three species using ggplot.Does it look like there are differences in the mean petal lengths?

```
# creates a boxplot of petal length for each species
iris %>%
  ggplot(aes(x=Species,y=Petal.Length)) +
    geom_boxplot()
```



It does looks like the mean petal lengths for each species are different, especially for setosa. Versicolor and virginica have means that are much closer to each other.

b. Create a linear model where sepal length is modeled by species. Give it an appropriate name.

```
# creates a linear model for sepal length based on species called SepalModel
SepalModel <- lm(Sepal.Length~Species-1, data=iris)
```

c. Type anova(your model name) in a code chunk.

```
# Uses ANOVA on SepalModel
anova(SepalModel)
```

```
## Analysis of Variance Table
```

```
##
## Response: Sepal.Length
##            Df Sum Sq Mean Sq F value    Pr(>F)
## Species     3 5184.9 1728.30  6521.7 < 2.2e-16 ***
## Residuals 147   39.0    0.27
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

d. What is the p-value for the test? What do you conclude.

The p-value is very close to zero, which is less than 0.05. We reject the null. There is sufficient evidence to suggest that the means for sepal length based on species are different.

e. Type summary(your model name) in a code chunk.

```r
# uses summary function on SepalModel
summary(SepalModel)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Species - 1, data = iris)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6880 -0.3285 -0.0060  0.3120  1.3120
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## Speciessetosa       5.0060     0.0728   68.76   <2e-16 ***
## Speciesversicolor   5.9360     0.0728   81.54   <2e-16 ***
## Speciesvirginica    6.5880     0.0728   90.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5148 on 147 degrees of freedom
## Multiple R-squared:  0.9925, Adjusted R-squared:  0.9924
## F-statistic:  6522 on 3 and 147 DF,  p-value: < 2.2e-16
```

f. What is the mean petal length for the species setosa?

From the summary table above, the mean sepal length for setosa is 5.0060.

g. What is the mean petal length for the species versicolor?

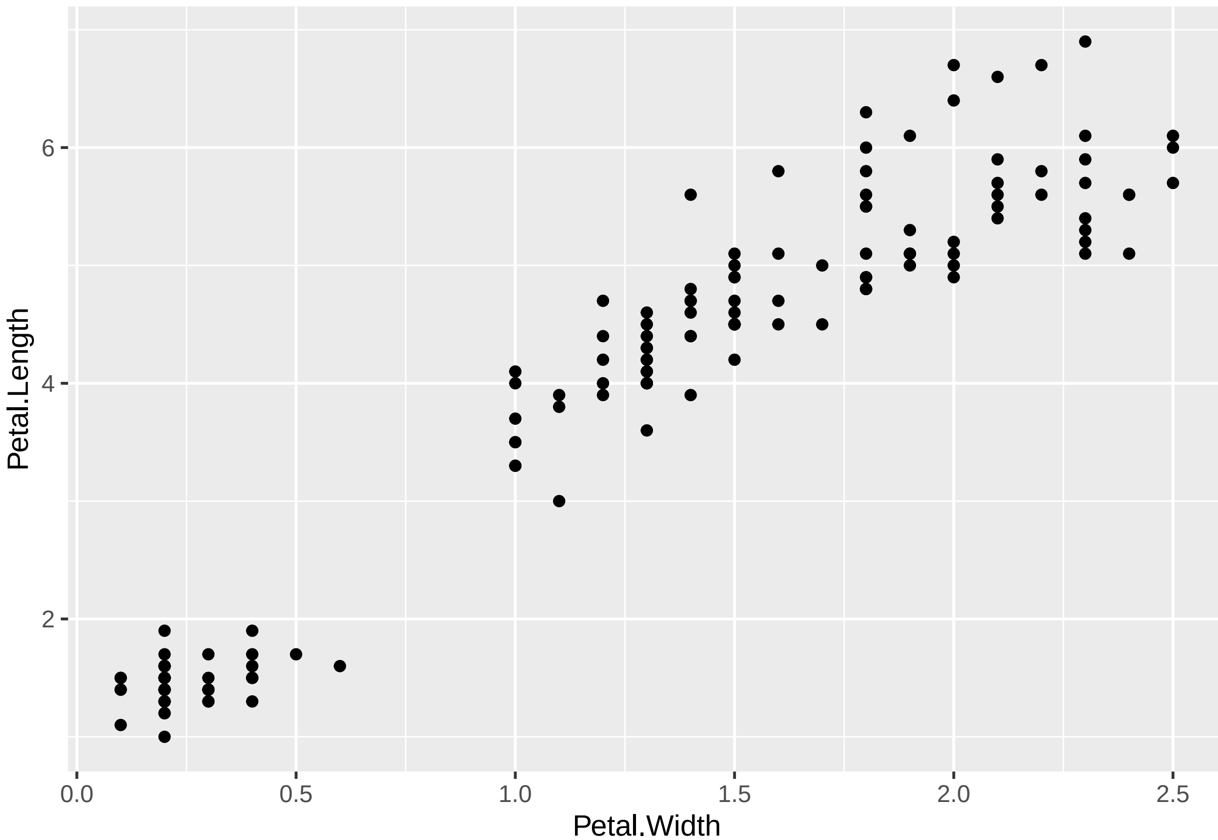
From the summary table above, the mean sepal length for versicolor is 5.9360.

Problem 3: Regression

Can we describe the relationship between petal length and petal width?

a. Create a scatterplot with petal length on the y-axis and petal width on the x-axis using ggplot.

```r
# creates a scatter plot comparing petal width and length in the iris data
iris %>%
  ggplot(aes(x=Petal.Width,y=Petal.Length)) +
  geom_point()
```



b. Create a linear model to model petal length with petal width (length is the response variable and width is the explanatory variable) using lm.

```
# creates a linear model between petal length and width from the iris data called PetalModel
PetalModel <- lm(Petal.Length~Petal.Width, data=iris)

# shows the PetalModel parameters
PetalModel
```

```
##
## Call:
## lm(formula = Petal.Length ~ Petal.Width, data = iris)
##
## Coefficients:
## (Intercept)  Petal.Width
##       1.084        2.230
```

c. What is the estimate of the slope parameter?

From the result above, the estimate for the slope parameter is 2.230.

d. What is the estimate of the intercept parameter?

From the result above, the estimate for the intercept parameter is 1.084.

e. Use summary() to get additional information.

```
# uses summary function on PetalModel
summary(PetalModel)
```

```
##
```

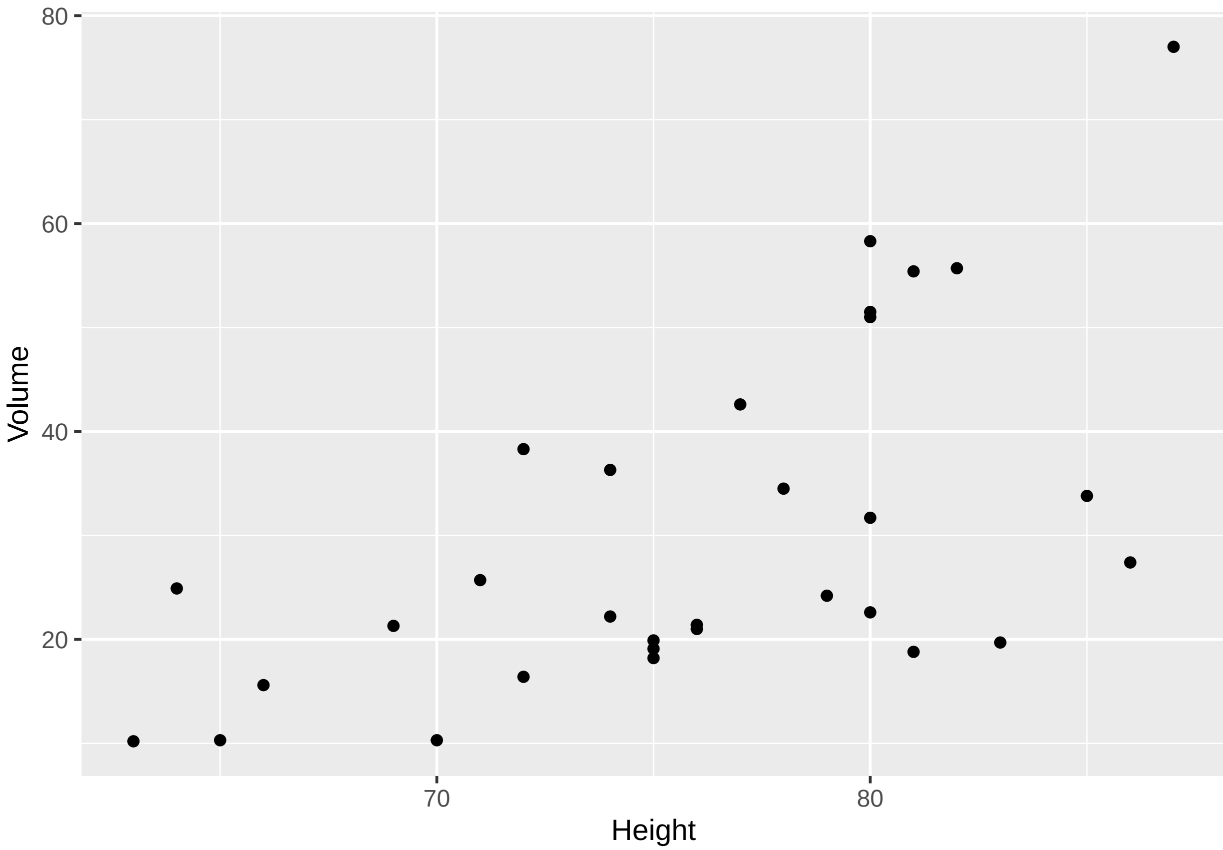
```
## Call:
## lm(formula = Petal.Length ~ Petal.Width, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33542 -0.30347 -0.02955  0.25776  1.39453
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.08356    0.07297   14.85   <2e-16 ***
## Petal.Width  2.22994    0.05140   43.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4782 on 148 degrees of freedom
## Multiple R-squared:  0.9271, Adjusted R-squared:  0.9266
## F-statistic:  1882 on 1 and 148 DF,  p-value: < 2.2e-16
```

Problem 4: Modeling Trees

Using the `trees` data frame that comes pre-installed in R, follow the steps below to fit the regression model that uses the tree `Height` to explain the `Volume` of wood harvested from the tree.

a. Create a scatterplot of the data using ggplot.

```r
# load in the trees data
data("trees")

# creates a plot of tree height vs tree volume
ggplot(data=trees,aes(x=Height,y=Volume)) +
  geom_point()
```



b.

Fit a `lm` model using the command `model <- lm(Volume ~ Height, data=trees)`.

```
# Creates linear model between volume and height for trees called TreeModel
TreeModel <- lm(Volume ~ Height, data=trees)

# shows the parameters for TreeModel
TreeModel
```

```
##
## Call:
## lm(formula = Volume ~ Height, data = trees)
##
## Coefficients:
## (Intercept)       Height
##     -87.124        1.543
```

c. Print out the table of coefficients with estimate names, estimated value, standard error, and upper and lower 95% confidence intervals.

```
# creates the prediction table with estimated values, standard error, and confidence
# intervals for TreeModel called TreeModelPred
TreeModelPred <- predict(TreeModel, se.fit = T, interval="confidence", level = .95)

# shows TreeModelPred
TreeModelPred
```

```
## $fit
##          fit       lwr      upr
## 1   20.91087 14.098550 27.72319
## 2   13.19412  3.254288 23.13395
```

```
## 3  10.10742 -1.223363 21.43821
## 4  23.99757 18.159758 29.83538
## 5  37.88772 31.592680 44.18275
## 6  40.97442 33.597379 48.35145
## 7  14.73747  5.471607 24.00333
## 8  28.62762 23.644217 33.61102
## 9  36.34437 30.506556 42.18218
## 10 28.62762 23.644217 33.61102
## 11 34.80102 29.345254 40.25678
## 12 30.17097 25.249799 35.09214
## 13 30.17097 25.249799 35.09214
## 14 19.36752 11.990482 26.74456
## 15 28.62762 23.644217 33.61102
## 16 27.08427 21.918668 32.24987
## 17 44.06112 35.450370 52.67186
## 18 45.60447 36.338602 54.87033
## 19 22.45422 16.159183 28.74926
## 20 11.65077  1.021703 22.27984
## 21 33.25767 28.092067 38.42327
## 22 36.34437 30.506556 42.18218
## 23 27.08427 21.918668 32.24987
## 24 23.99757 18.159758 29.83538
## 25 31.71432 26.730917 36.69772
## 26 37.88772 31.592680 44.18275
## 27 39.43107 32.618747 46.24339
## 28 36.34437 30.506556 42.18218
## 29 36.34437 30.506556 42.18218
## 30 36.34437 30.506556 42.18218
## 31 47.14782 37.207982 57.08765
##
## $se.fit
##  [1] 3.330833 4.860008 5.540104 2.854354 3.077912 3.606948 4.530476 2.436597
##  [9] 2.854354 2.436597 2.667555 2.406169 2.406169 3.606948 2.436597 2.525682
## [17] 4.210161 4.530476 3.077912 5.197005 2.525682 2.854354 2.525682 2.854354
## [25] 2.436597 3.077912 3.330833 2.854354 2.854354 2.854354 4.860008
##
## $df
## [1] 29
##
## $residual.scale
## [1] 13.39698
```

d.Add the model fitted values to the `trees` data frame along with the regression model confidence intervals. Note: the book does this in a super convoluted way. Don't follow the model in the book. Instead try cbind.

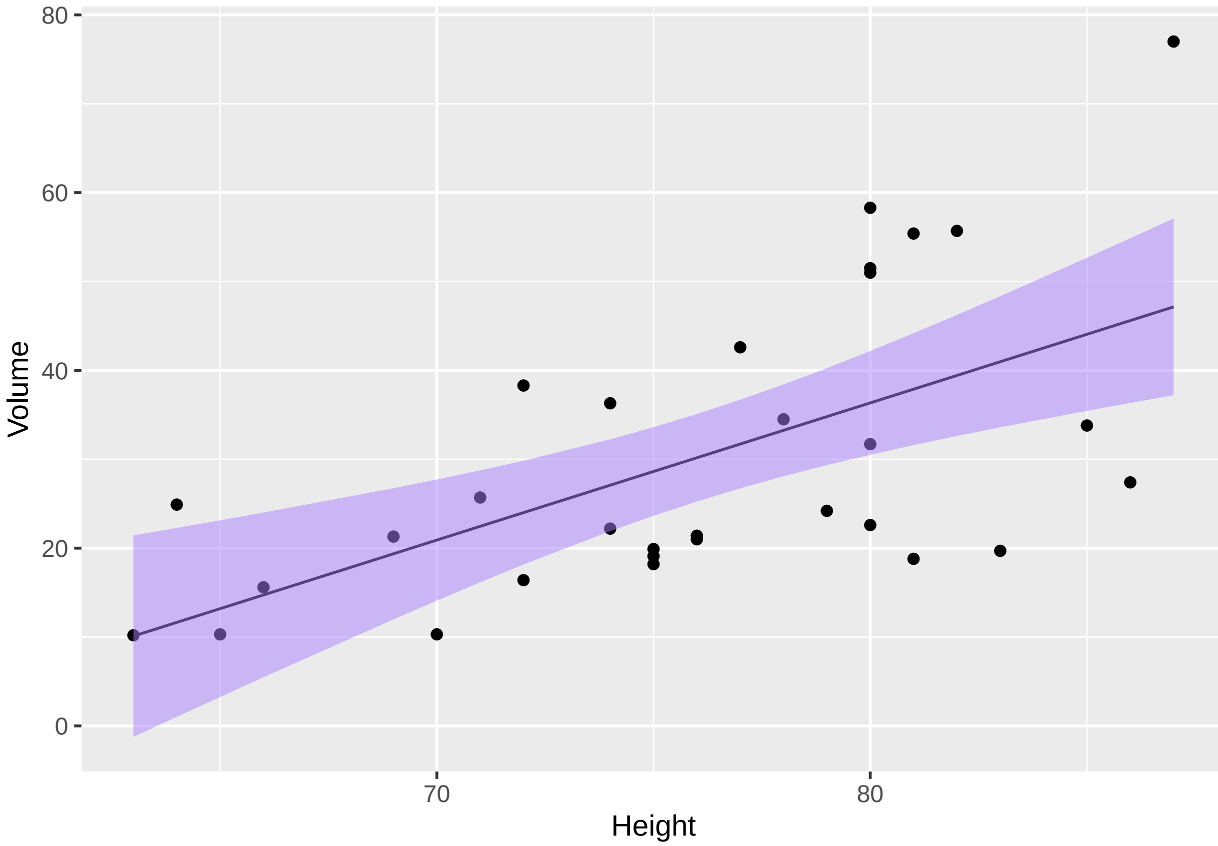
```
# adds the model fitted values and confidence intervals to trees data and saves it under
# TreeModelGraph
TreeModelGraph <- cbind(trees, predict(TreeModel, interval="confidence"))

# shows TreeModelGraph
TreeModelGraph
```

```
##    Girth Height Volume      fit       lwr      upr
## 1    8.3     70   10.3 20.91087 14.098550 27.72319
## 2    8.6     65   10.3 13.19412  3.254288 23.13395
```

```
## 3    8.8     63   10.2 10.10742 -1.223363 21.43821
## 4   10.5     72   16.4 23.99757 18.159758 29.83538
## 5   10.7     81   18.8 37.88772 31.592680 44.18275
## 6   10.8     83   19.7 40.97442 33.597379 48.35145
## 7   11.0     66   15.6 14.73747  5.471607 24.00333
## 8   11.0     75   18.2 28.62762 23.644217 33.61102
## 9   11.1     80   22.6 36.34437 30.506556 42.18218
## 10  11.2     75   19.9 28.62762 23.644217 33.61102
## 11  11.3     79   24.2 34.80102 29.345254 40.25678
## 12  11.4     76   21.0 30.17097 25.249799 35.09214
## 13  11.4     76   21.4 30.17097 25.249799 35.09214
## 14  11.7     69   21.3 19.36752 11.990482 26.74456
## 15  12.0     75   19.1 28.62762 23.644217 33.61102
## 16  12.9     74   22.2 27.08427 21.918668 32.24987
## 17  12.9     85   33.8 44.06112 35.450370 52.67186
## 18  13.3     86   27.4 45.60447 36.338602 54.87033
## 19  13.7     71   25.7 22.45422 16.159183 28.74926
## 20  13.8     64   24.9 11.65077  1.021703 22.27984
## 21  14.0     78   34.5 33.25767 28.092067 38.42327
## 22  14.2     80   31.7 36.34437 30.506556 42.18218
## 23  14.5     74   36.3 27.08427 21.918668 32.24987
## 24  16.0     72   38.3 23.99757 18.159758 29.83538
## 25  16.3     77   42.6 31.71432 26.730917 36.69772
## 26  17.3     81   55.4 37.88772 31.592680 44.18275
## 27  17.5     82   55.7 39.43107 32.618747 46.24339
## 28  17.9     80   58.3 36.34437 30.506556 42.18218
## 29  18.0     80   51.5 36.34437 30.506556 42.18218
## 30  18.0     80   51.0 36.34437 30.506556 42.18218
## 31  20.6     87   77.0 47.14782 37.207982 57.08765
```

e. Graph the data and fitted regression line and uncertainty ribbon.

```r
# graphs the tree data with the fitted regression line and uncertainty ribbon
ggplot(data=TreeModelGraph,aes(x=Height,y=Volume)) +
  geom_point() +
  geom_line(aes(y=fit)) +
  geom_ribbon(aes(ymin=lwr, ymax=upr), alpha=0.5, fill = "mediumpurple1")
```



f. Add the R-squared value as an annotation to the graph using annotate.

```r
# shows summary of TreeModel to find R-squared
summary(TreeModel)
```

```
##
## Call:
## lm(formula = Volume ~ Height, data = trees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.274  -9.894  -2.894  12.068  29.852
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -87.1236    29.2731  -2.976 0.005835 **
## Height        1.5433     0.3839   4.021 0.000378 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.4 on 29 degrees of freedom
## Multiple R-squared:  0.3579, Adjusted R-squared:  0.3358
## F-statistic: 16.16 on 1 and 29 DF,  p-value: 0.0003784
```

```r
# adds R-squared value to graph
ggplot(data=TreeModelGraph,aes(x=Height,y=Volume)) +
  geom_point() +
  geom_line(aes(y=fit)) +
```

```r
geom_ribbon(aes(ymin=lwr, ymax=upr), alpha=0.5, fill = "mediumpurple1") +
annotate("text", x=67, y=70, label="R-squared value = 0.3358")
```