W03D

Benjamin Huang

2024-09-12

#trees Model Summary

Below are histograms for each variable in the built-in data set trees.

##Diameter, Height and Volume for Black Cherry Trees

###Description

This data set provides measurements of the diameter, height and volume of timber in 31 felled black cherry trees. Note that the diameter (in inches) is erroneously labelled Girth in the data. It is measured at 4 ft 6 in above the ground.

###Usage

trees

###Format

A data frame with 31 observations on 3 variables.

[,1] Girth; numeric; tree diameter (rather than girth, actually) in inches

[,2] Height; numeric; height in ft

[,3] Volume; numeric; volume of timber in cubic ft

```
#install.packages("tidyverse")
#install.packages("gridExtra")
#install.packages("GGally")
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## Warning: package 'tibble' was built under R version 4.2.3
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'purrr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'stringr' was built under R version 4.2.3

## Warning: package 'forcats' was built under R version 4.2.3

## Warning: package 'lubridate' was built under R version 4.2.3

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.2.3

##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.2.3

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

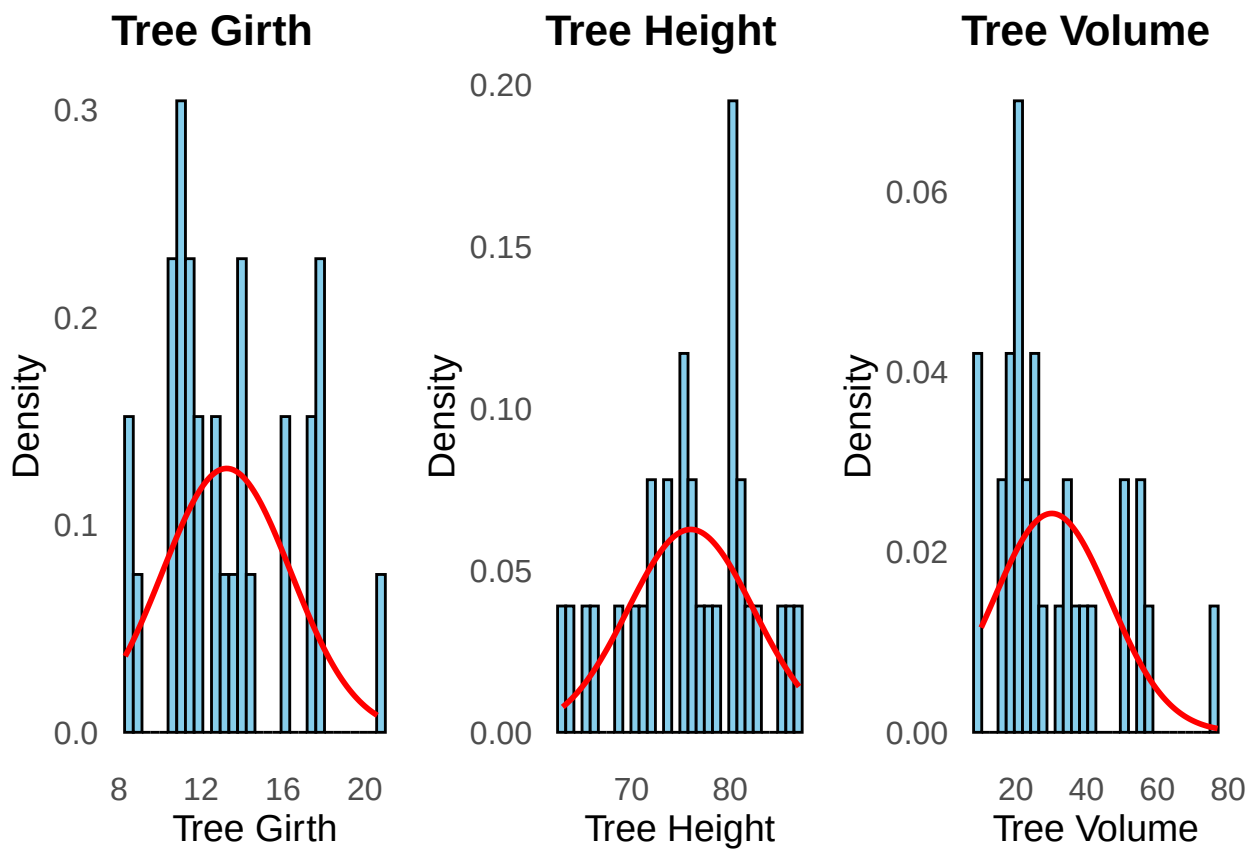
```
# Function to plot histograms with normal distribution overlay and no grid
plot_histogram_with_normal <- function(data, column_name, title)  {
  mean_val <- mean(data[[column_name]])
  sd_val <- sd(data[[column_name]])

  ggplot(data, aes(x = .data[[column_name]])) +
    geom_histogram(aes(y = ..density..), bins = 30, color = "black", fill = "skyblue") +
    stat_function(fun = dnorm, args = list(mean = mean_val, sd = sd_val), color = "red", size = 1) +
    ggtitle(title) +
    xlab(title) +
    ylab("Density") +
    theme_minimal() +
    theme(
      plot.title = element_text(size = 16, face = "bold"),
      axis.title.x = element_text(size = 14),
```

```
      axis.title.y = element_text(size = 14),
      axis.text = element_text(size = 12),
      panel.grid = element_blank() # Removes gridlines
    )
}

# Create individual plots
p1 <- plot_histogram_with_normal(trees, "Girth", "Tree Girth")
```

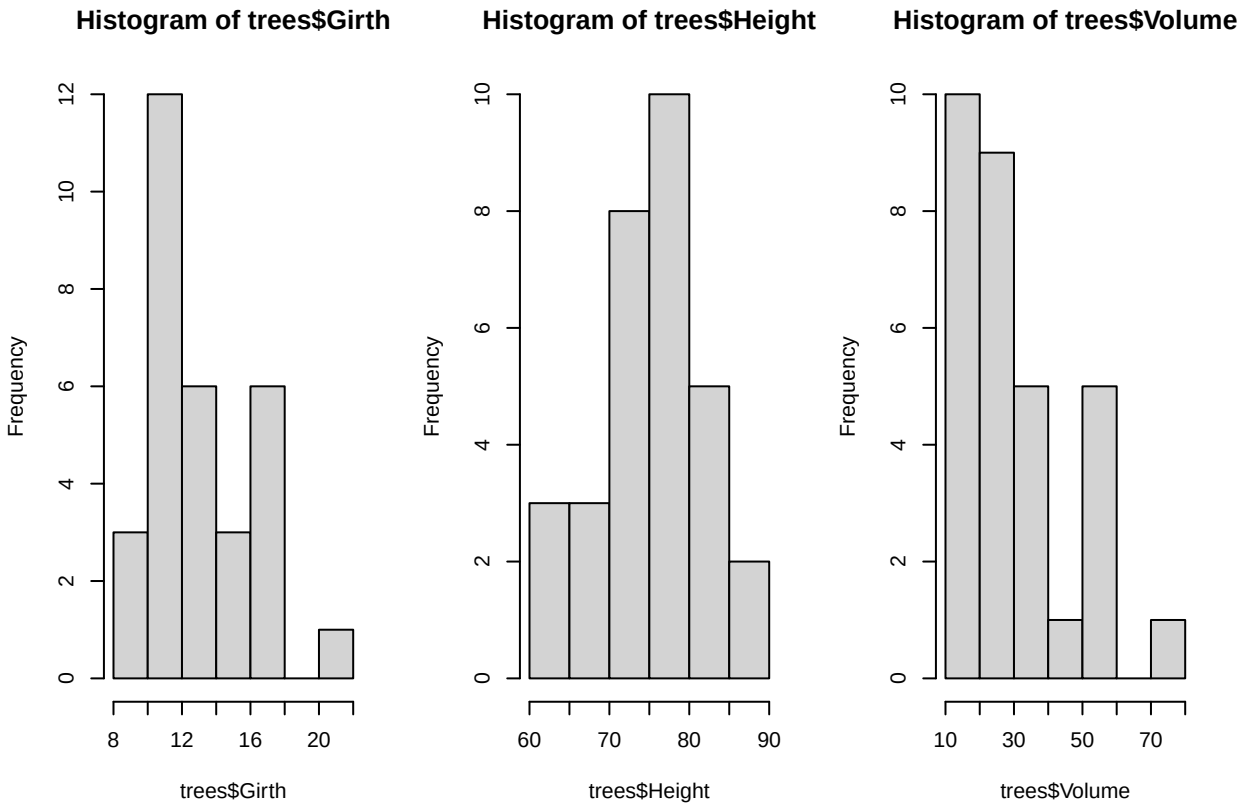
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
p2 <- plot_histogram_with_normal(trees, "Height", "Tree Height")
p3 <- plot_histogram_with_normal(trees, "Volume", "Tree Volume")

# Arrange them in one row
library(gridExtra)
grid.arrange(p1, p2, p3, nrow = 1)
```

```
## Warning: The dot-dot notation (‘..density..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(density)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```
par(mfrow = c(1, 3))
hist(trees$Girth); hist(trees$Height); hist(trees$Volume)
```

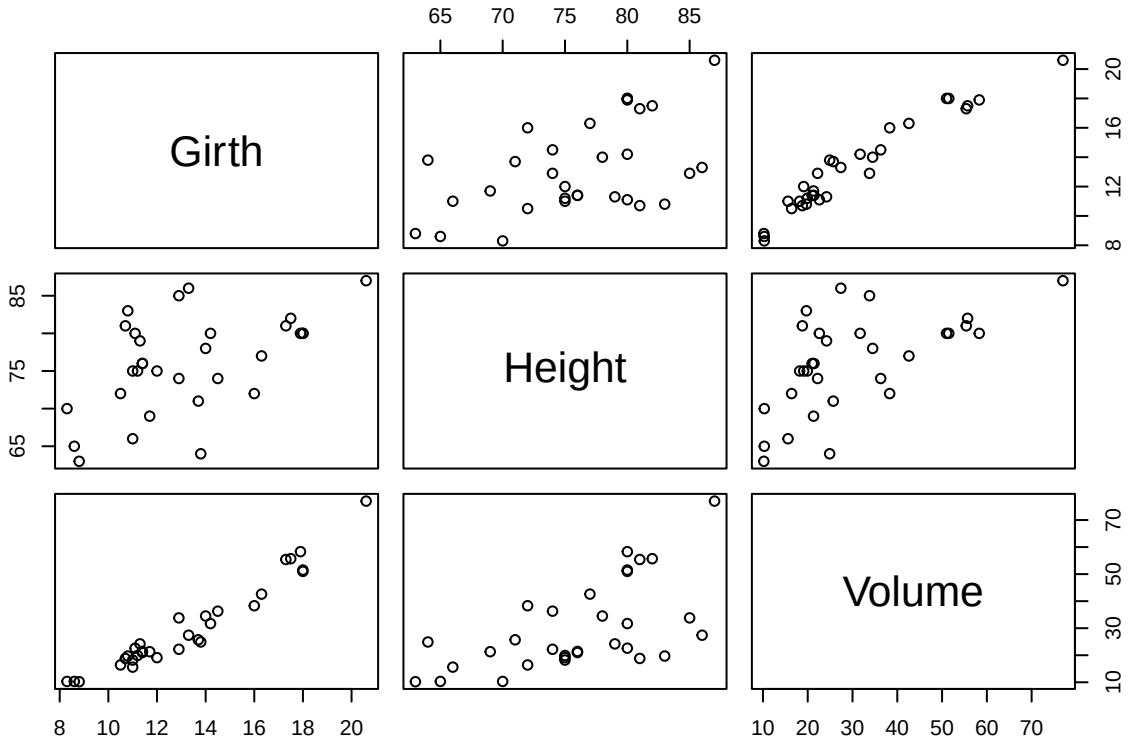


Q1. Which variable is closest to being normally distributed?

Answer: None of them

Next are plots of the columns against each other.

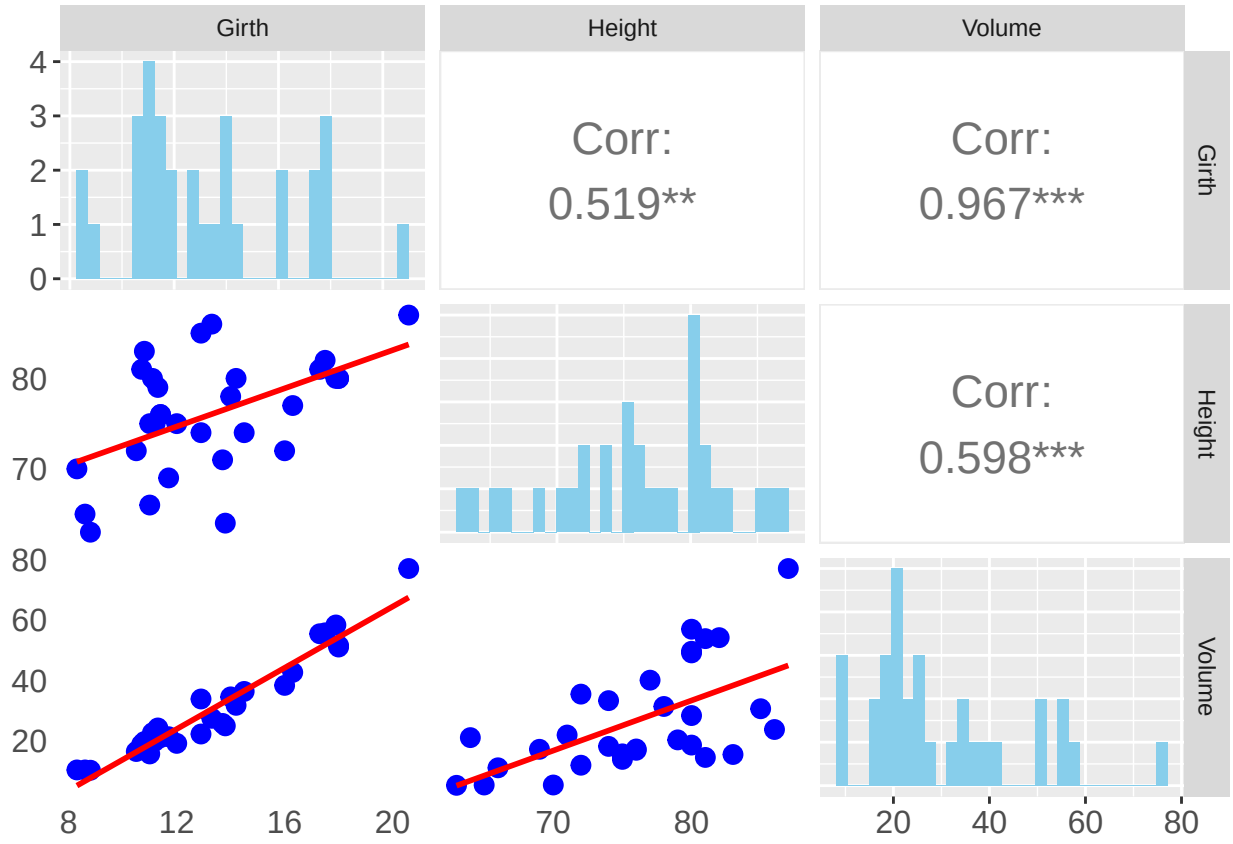
```
plot(trees)
```




```
# Function to add a custom plot for scatter plots with correlation and best-fit line
custom_plot <- function(data, mapping, ...)  {
  ggplot(data = data, mapping = mapping) +
    geom_point(color = "blue", size = 3) + # Scatter plot points
    geom_smooth(method = "lm", color = "red", se = FALSE) + # Best-fit
    theme_minimal() +
    theme(
      panel.grid = element_blank() # Remove grid lines
    )
}

# Create the ggpairs matrix plot with correlation and best-fit line
ggpairs(trees,
        lower = list(continuous = custom_plot), # Use custom scatter plots in lower triangle
        diag = list(continuous = wrap("barDiag", fill = "skyblue")), # Histograms on diagonal
        upper = list(continuous = wrap("cor", size = 6)) # Correlation coefficient in upper triangle
        ) +
  theme(
    plot.title = element_text(size = 16, face = "bold"),
    axis.title.x = element_text(size = 14),
    axis.title.y = element_text(size = 14),
    axis.text = element_text(size = 12)
  )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `geom_smooth()` using formula = 'y ~ x'
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Q2. Which pair has the highest correlation?

Answer: Girth and Volume

Below is a summary for a regression using Volume as the y and Girth as the x.

Note the "t value and"Pr(>|t|)" columns. In lecture, we called these the t-ratio and the p-value. The null hypothesis is that the coefficient is 0.

Next the these columns are three asterisks, ***. In the row below, you can see the significance level that these coefficients exceed. Both exceed a 0.1% significance level, giving confidence that neither is zero.

```
lm <- lm(Volume~Girth, data = trees)
summary(lm)
```

```
##
## Call:
## lm(formula = Volume ~ Girth, data = trees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.065 -3.107  0.152  3.495  9.587
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
## Girth         5.0659     0.2474   20.48  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.252 on 29 degrees of freedom
## Multiple R-squared:  0.9353, Adjusted R-squared:  0.9331
## F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16
```

Q3. What is the equation of the fitted regression line?

Answer:

$Volume = 5.0659 \times Girth - 36.9435$

Q4. How is the "t value column calculated?

Answer:

$\frac{Coefficient\ Estimate}{Std.\ Error}$

Q5. Explain what "< 2e-16" means in the output above.

Answer:

This means the value (a p-value in this case), is less that 2*10^(-16) = 0.000...0002. A very small value, close to zero.

Let's compare this regression to a regression of Volume on Height, which is below.

Note that the Intercept loses significance in this regression.

```
lm2 <- lm(Volume~Height, data = trees)
summary(lm2)
```

```
##
## 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
```

Q6. What is the p-value corresponding to $H_o : \beta_1 = 0$? Answer: $p = 0.000378$

In the code lm2 <- lm(Volume ~ Height, data = trees) followed by summary(lm2), the p-value corresponding to the hypothesis test:

$H_0 : \beta_1 = 0 \, vs. \, H_1 : \beta_1 \neq 0$

(i.e., testing whether the slope of Height is significantly different from zero) can be found under the column labeled Pr(>|t|) in the output of summary(lm2). This p-value is located in the row corresponding to Height. It tests the null is 0. If the p-value is small (typically less than 0.05, but other values are also accepted), it indicates that you should reject the null hypothesis and conclude that Height is a significant predictor of Volume.

We can find the percentiles from a t-distribution using the qt(p, df) function.

The pt(t, df) function gives the corresponding probability for a t-value.

```
qt(0.005835 / 2, 29) #quantile of the t distribution
```

```
## [1] -2.97621
```

```
pt(-2.976, 29)*2 #probability level of the t distribution
```

```
## [1] 0.005838043
```

Q7. Explain why there is "/2" in qt and "*2" outside pt.

Answer: This is because we are doing a two-tailed test, which is represented by Pr(>|t|).

I don't understand what this question is trying to achieve, maybe we could as a different question, whether the previous t test corresponds to a one-sided or a double-sided hypothesis test(answer, double sided, hint from the columns in the summary object)

Proposed Question and Revision

Two-Sided Hypothesis Testing

A two-sided hypothesis test assesses whether a parameter is significantly different from a null value in either direction-either higher or lower. For example, when testing if a regression coefficient β_1 is different from zero, the null hypothesis (H_0) might be $\beta_1 = 0$, while the alternative hypothesis (H_A) would be $\beta_1 \neq 0$. The test checks for evidence that β_1 is either significantly greater than or less than zero.

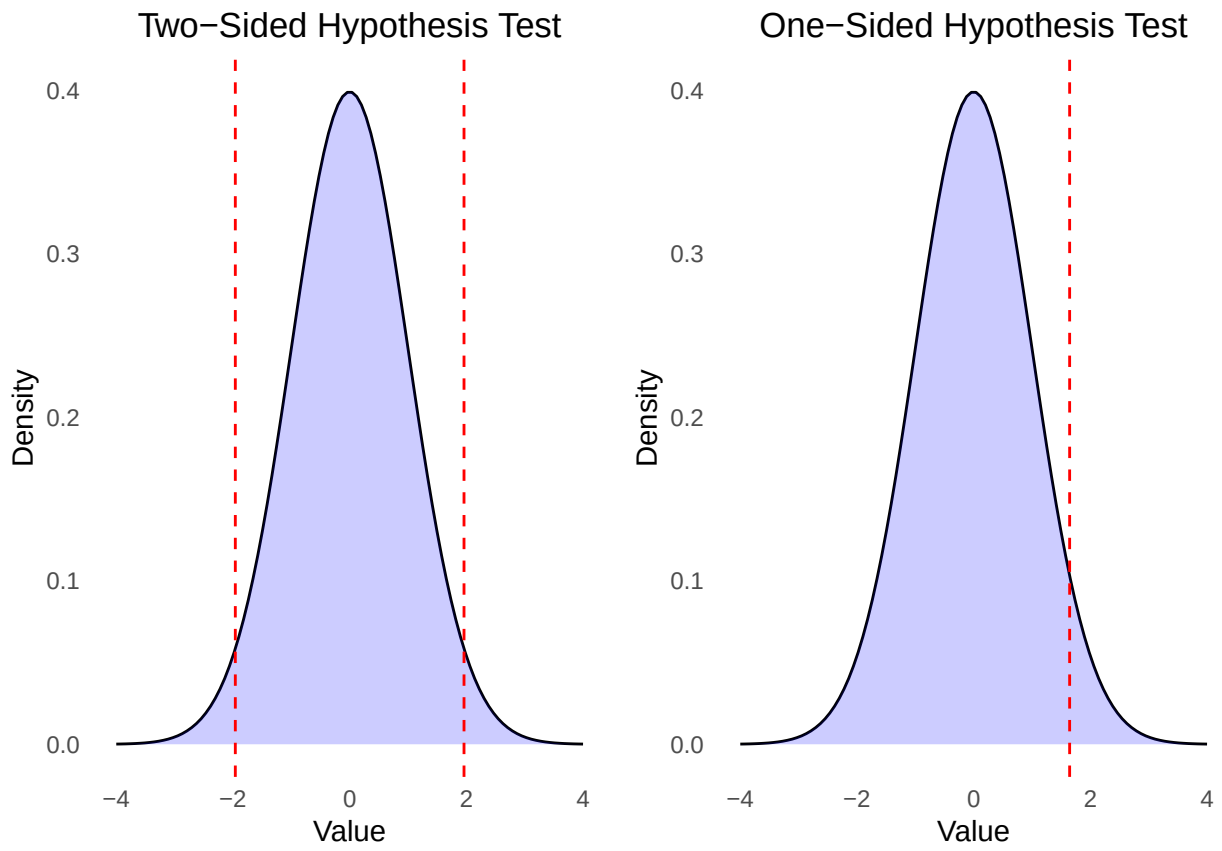
To make this more concrete, consider a two-sided confidence interval for β_1:

$\hat{\beta}_1 \pm t_{\alpha/2, n-2} \times SE(\hat{\beta}_1)$ where:

1. $\hat{\beta}_1$ is the estimated regression coefficient.
2. $t_{\alpha/2, n-2}$ is the critical value from the t-distribution corresponding to the $\alpha/2$ level of significance with $n - 2$ degrees of freedom.
3. $SE(\hat{\beta}_1)$ is the standard error of the estimated coefficient.

The confidence interval provides a range where the true value of β_1 is likely to fall with a specified level of confidence. If zero falls outside this interval, it suggests that β_1 is significantly different from zero in either direction.

```r
# Define parameters for the plots
mean_val <- 0
sd_val <- 1
alpha <- 0.05
crit_val <- qnorm(1 - alpha/2) # Two-sided critical value
crit_val_one_sided <- qnorm(1 - alpha) # One-sided critical value

# Create data frame fir plotting
x <- seq(-4, 4, length.out = 100)
y <- dnorm(x, mean = mean_val, sd = sd_val)
data <- data.frame(x = x, y = y)

# Plot for Two-Sided Hypothesis Test
plot_two_sided <- ggplot(data, aes(x = x, y = y)) +
  geom_line() +
  geom_ribbon(aes(ymin = 0, ymax = y), fill = "blue", alpha = 0.2) +
  geom_vline(xintercept = c(-crit_val, crit_val), linetype = "dashed", color = "red") +
  ggtitle("Two-Sided Hypothesis Test") +
  xlab("Value") +
  ylab("Density") +
  theme_minimal() +
  theme(panel.grid.major = element_blank(), # Remove major grid lines
        panel.grid.minor = element_blank(), # Remove minor grid lines
        plot.title = element_text(hjust = 0.5))

# Plot for One-Sided Hypothesis Test
plot_one_sided <- ggplot(data, aes(x = x, y = y)) +
  geom_line() +
  geom_ribbon(aes(ymin = 0, ymax = y), fill = "blue", alpha = 0.2) +
  geom_vline(xintercept = crit_val_one_sided, linetype = "dashed", color = "red") +
  ggtitle("One-Sided Hypothesis Test") +
  xlab("Value") +
  ylab("Density") +
  theme_minimal() +
  theme(panel.grid.major = element_blank(), # Remove major grid lines
        panel.grid.minor = element_blank(), # Remove minor grid lines
        plot.title = element_text(hjust = 0.5))

# Arrange plots side by side
grid.arrange(plot_two_sided, plot_one_sided, ncol = 2)
```



Q8. Find the approximate and actual 95% confidence interval for β_1 for lm2.

```
#Answer
#Approximation
1.5433 - 2*0.3839; 1.5433 +2*0.3839
```

```
## [1] 0.7755
```

```
## [1] 2.3111
```

```
#Actual
1.5433 + qt(.025, 29)*0.3839; 1.5433 + qt(.975, 29)*0.3839
```

```
## [1] 0.7581363
```

```
## [1] 2.328464
```

The 95% confidence interval for a regression coefficient β_1 provides a range that likely contains the true coefficient value based on the sample data. The approximate confidence interval is calculated as:

$\hat{\beta}_1 \pm 2 \times SE(\hat{\beta}_1)$

The factor of 2 is a rough approximation for the critical value from the t-distribution when the sample size is large (around 30 observations or more).

The actual 95% confidence interval is more preciesly calculated using:

$\hat{\beta}_1 \pm t_{\alpha/2,n-2} \times SE(\hat{\beta}_1)$

where:

1, $\hat{\beta}_1 = 1.5433$

2, $t_{\alpha/2,n-2}$ is the critical t-value for a 95% confidence level with n-2 degrees of freedom.

3, $SE(\hat{\beta}_1) = 0.3839$.

where actual values from the t distribution are used. This gives a more accurate interval than the approximation, reflecting the t-distribution's dependence on sample size.

Bonus Question: Which one is wider? Why?

This can also be done with the confint() function.

```
confint(lm2, "Height", level = 0.95) #confidence interval
```

```
##           2.5 %   97.5 %
## Height 0.758249 2.328451
```

Q9. Find the 99.9% confidnece interval for the intercept.

```
#Answer
confint(lm2, "(Intercept)", level = 0.999)
```

```
##                0.05 % 99.95 %
## (Intercept) -194.2458 19.9986
```

A prediction interval in regression analysis estimates the range within which a new individual observation is expected to fall, considering both the uncertainty in the regression model and the natural variability in the data around the regression line. It provides a broader range compared to a confidence interval because it includes the variability of individual data points as well as the uncertainty in estimating the model parameters. In contrast, a confidence interval estimates the range within which the true population parameter (e.g., the slope of the regression line) lies, reflecting only the precision of the parameter estimates and not the variability of individual observations. Thus, while a prediction interval gives a wider range for a individual future observations, a confidence interval focuses on the accuracy of the model's parameter estimates.

Prediction Interval and Confidence Interval

The prediction interval for a new observation y_{new} at given value x_{new} is given by:

$\hat{y}_{new} \pm t_{\alpha/2,n-2} \cdot \sqrt{MSE(1 + \frac{1}{n} + \frac{(x_{new} - \bar{x})^2}{S_{xx}})}$

where:

$\hat{y}_{new}$ is the predicted value from the regression model. $t_{\alpha/2,n-2}$ is the critical value from the t-distribution with $n-2$ degrees of freedom. MSE is the mean squared error of the regression model. n is the number of observations. x_{new} is the new predictor value. $\bar{x}$ is the mean of the predictor values. $S_{xx} = \sum (x_i - \bar{x})^2$ is the sum of squared deviations of the predictor values.

The confidence interval for the regression coefficient β_1 is given by:

$\hat{\beta}_1 \pm t_{\alpha/2,n-2} \cdot SE(\hat{\beta}_1)$

where:

$\hat{\beta}_1$ is the estimated coefficient from the regression model. $t_{\alpha/2,n-2}$ is the critical value from the t-distribution with $n-2$ degrees of freedom. $SE(\hat{\beta}_1)$ is the standard error of the estimated coefficient.

Mathematical Difference Prediction Interval: Includes terms to account for both the uncertainty in predicting the mean response ($\hat{y}_{new}$) and the variability of the individual observations around the regression line. It incorporates the mean squared error and additional terms reflecting the variability around the regression line.

Confidence Interval: Focuses on the precision of the estimated regression coefficient $\hat{\beta}_1$ and does not include terms for the variability of individual observations. It reflects only the error in estimating the regression parameter.

In summary, while the prediction interval accounts for the variability of individual observations and uncertainty in predicting the mean response, the confidence interval only addresses the precision of the estimated regression coefficients.

To create a prediction interval, we need to define a new data set that contains value of Height.

This is done below.

```
newdata = data.frame(Height = mean(trees$Height))
predict.lm(lm2, newdata = newdata, interval = "prediction", level = .95)
```

```
##        fit      lwr     upr
## 1 30.17097 2.332636 58.0093
```

Create a new data frame with a single value for Height set to the mean height of the trees in the dataset. Uses the linear model lm2 to predict the response for this mean height, including a 95% prediction interval. This prediction interval estimates where a new individual observation of Volume is likely to fall when Height is at its average value.