STA445 - Homework 7

Richard McCormick

2023-11-06

1. The infmort data set from the package faraway gives the infant mortality rate for a variety of countries. The information is relatively out of date (from 1970s?), but will be fun to graph. Visualize the data using by creating scatter plots of mortality vs income while faceting using region and setting color by oil export status. Utilize a log10 transformation for both mortality and income axes. This can be done either by doing the transformation inside the aes() command or by utilizing the scale_x_log10() or scale_y_log10() layers. The critical difference is if the scales are on the original vs log transformed scale. Experiment with both and see which you prefer.

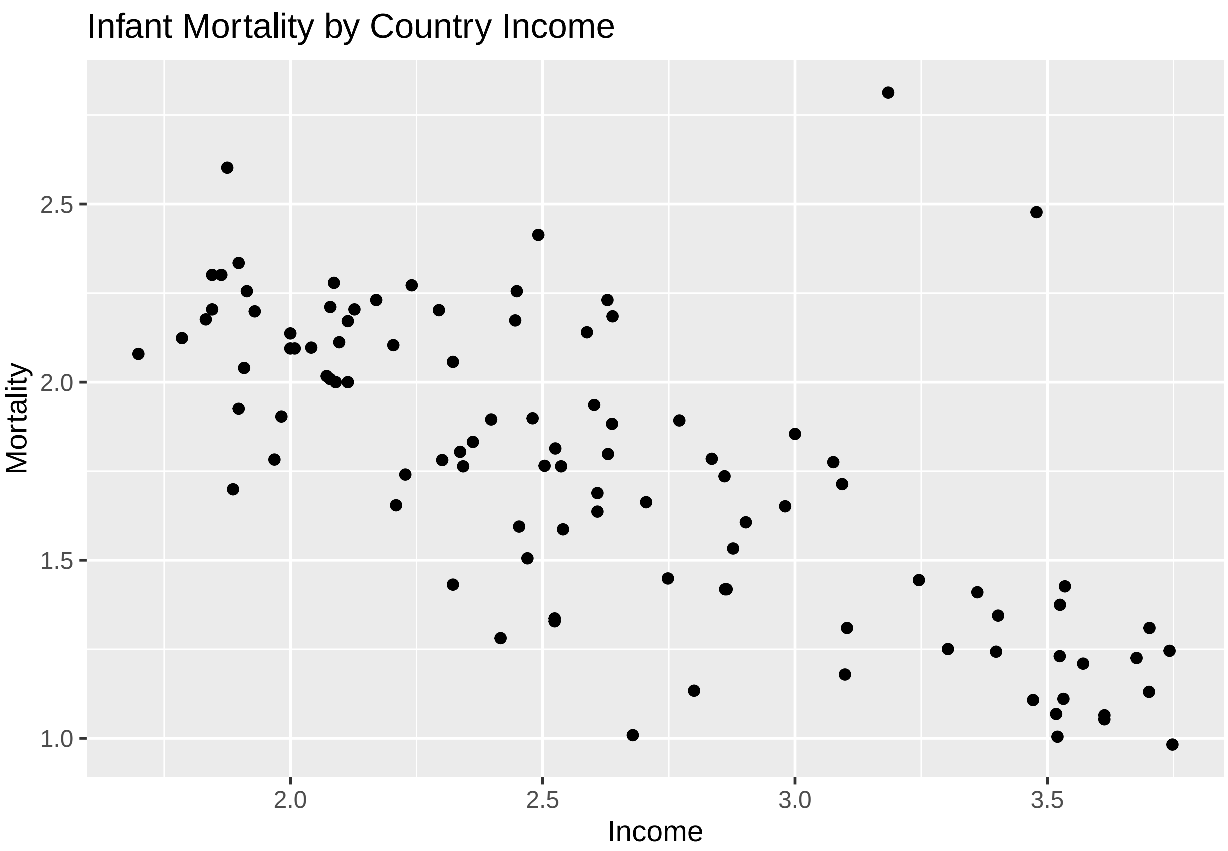
a. The rownames() of the table gives the country names and you should create a new column that contains the country names. *rownames

```
inf.data <- infmort
inf.data <- tibble::rownames_to_column( inf.data, "country" )
```

b. Create scatter plots with the log10() transformation inside the aes() command.

```
ggplot( data=inf.data, aes( x=log10( income ), y=log10( mortality ) ) ) +
  geom_point( ) +
  labs( title="Infant Mortality by Country Income",
        x="Income", y="Mortality" )
```

```
## Warning: Removed 4 rows containing missing values (‘geom_point()‘).
```

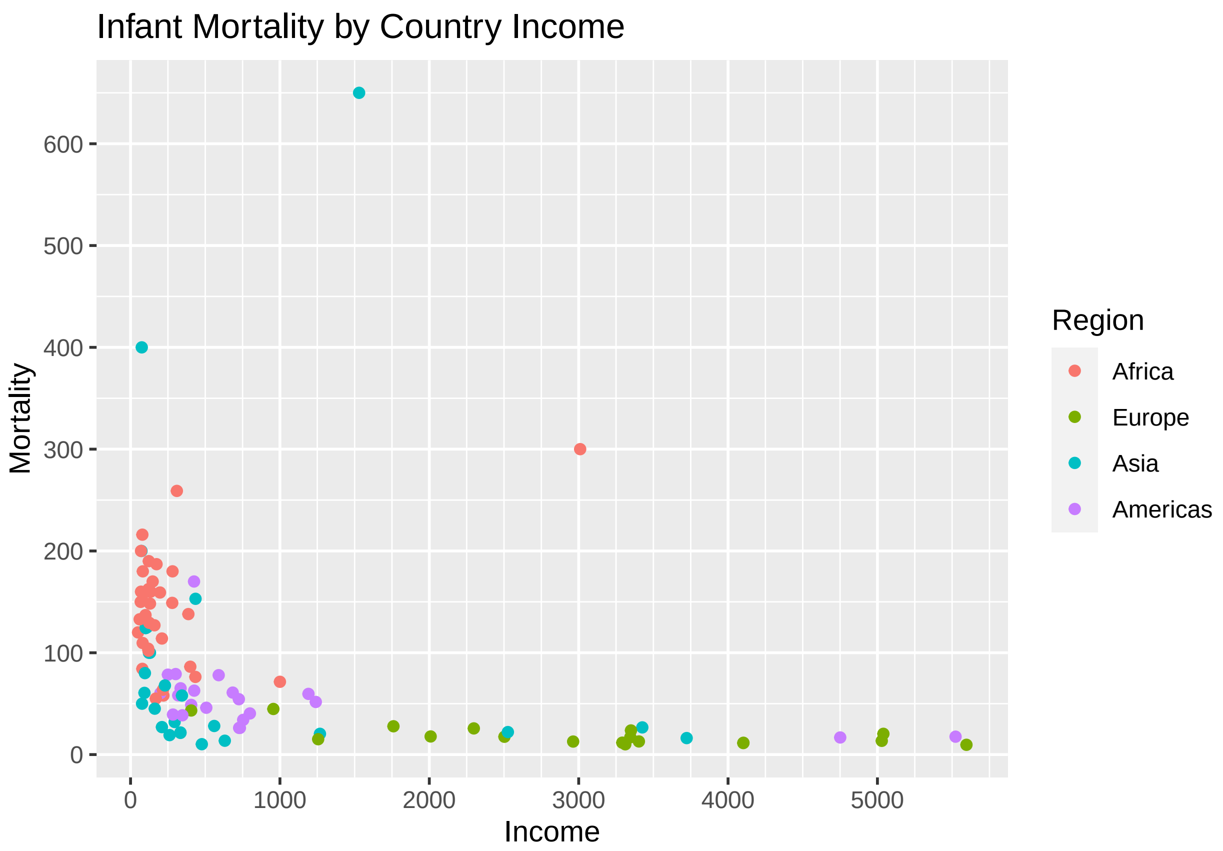


c. Create the scatter plots using the scale_x_log10() and scale_y_log10(). Set the major and minor breaks to be useful and aesthetically pleasing. Comment on which version you find easier to read.

```
ggplot( data=inf.data, aes( x=income, y=mortality ) ) +
  geom_point( aes( color=region ) ) +
  scale_x_log10() +
  scale_y_log10() +
  scale_x_continuous( breaks=seq( 0, 10000, by=1000 ),
                      minor_breaks=seq( 0, 10000, by=250 ) ) +
  scale_y_continuous( breaks=seq( 0, 1000, by=100 ),
                      minor_breaks=seq( 0, 1000, by=50 ) ) +
  labs( title="Infant Mortality by Country Income",
        x="Income", y="Mortality", color="Region" )
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```

```
## Warning: Removed 4 rows containing missing values (‘geom_point()‘).
```

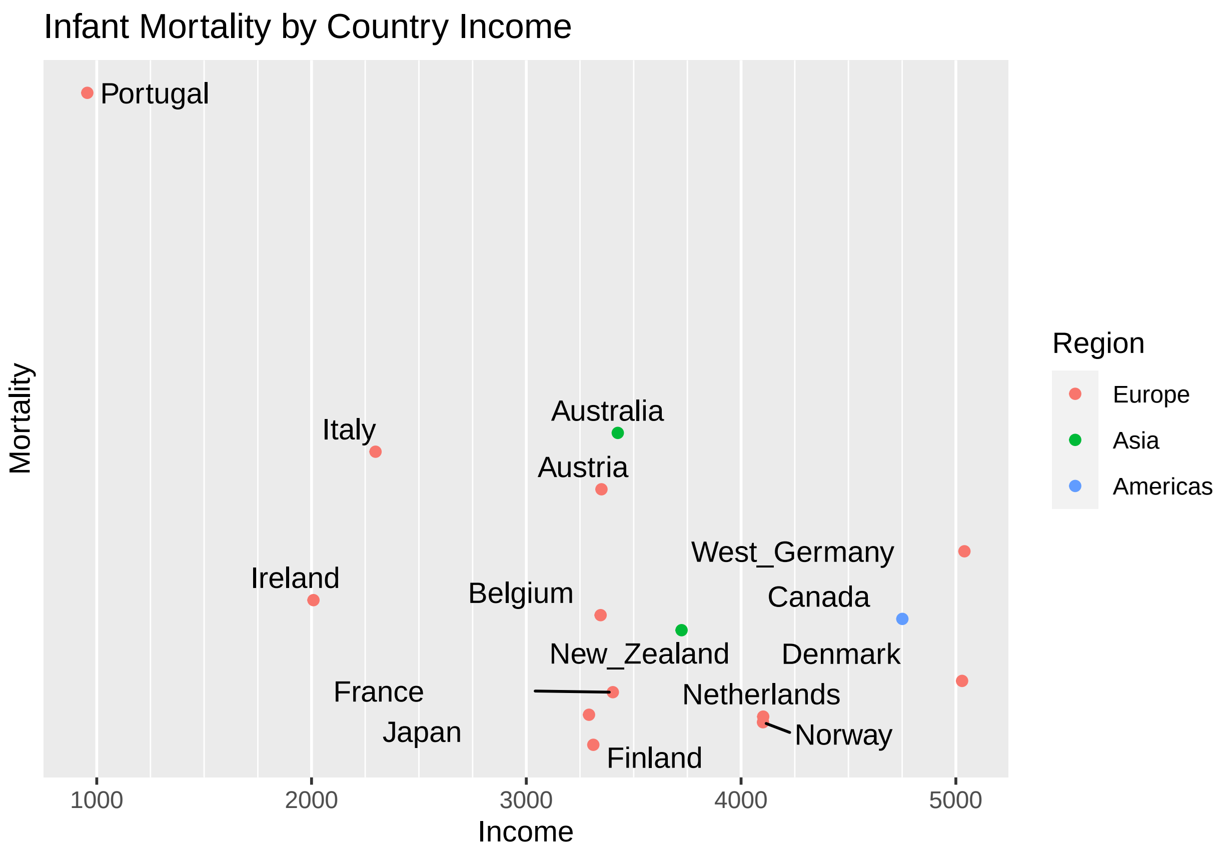


Having distinct continual major and minor breaks certainly makes the graph easier to read. This ensures that the axes are scaled at a continual interval that makes it much easier to interpret.

d. The package ggrepel contains functions geom_text_repel() and geom_label_repel() that mimic the basic geom_text() and geom_label() functions in ggplot2, but work to make sure the labels don't overlap. Select 10-15 countries to label and do so using the geom_text_repel() function.

```
ggplot( data=inf.data[1:15,], aes( x=income, y=mortality ) ) +
  geom_point( aes( color=region ) ) +
  scale_x_log10() +
  scale_y_log10() +
  scale_x_continuous( breaks=seq( 0, 10000, by=1000 ),
                      minor_breaks=seq( 0, 10000, by=250 ) ) +
  scale_y_continuous( breaks=seq( 0, 1000, by=100 ),
                      minor_breaks=seq( 0, 1000, by=50 ) ) +
  labs( title="Infant Mortality by Country Income",
        x="Income", y="Mortality", color="Region" ) +
  geom_text_repel( aes( label=country ) )
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```


Infant Mortality by Country Income
Mortality
Income
1000
2000
3000
4000
5000
Portugal
Italy
Australia
Austria
West_Germany
Ireland
Belgium
Canada
New_Zealand
Denmark
France
Netherlands
Japan
Norway
Finland
Region
Europe
Asia
Americas

2. Using the datasets::trees data, complete the following:

a. Create a regression model for y=Volume as a function of x=Height.

```
tree.data <- trees
tree.model <- lm( tree.data$Volume ~ tree.data$Height )
```

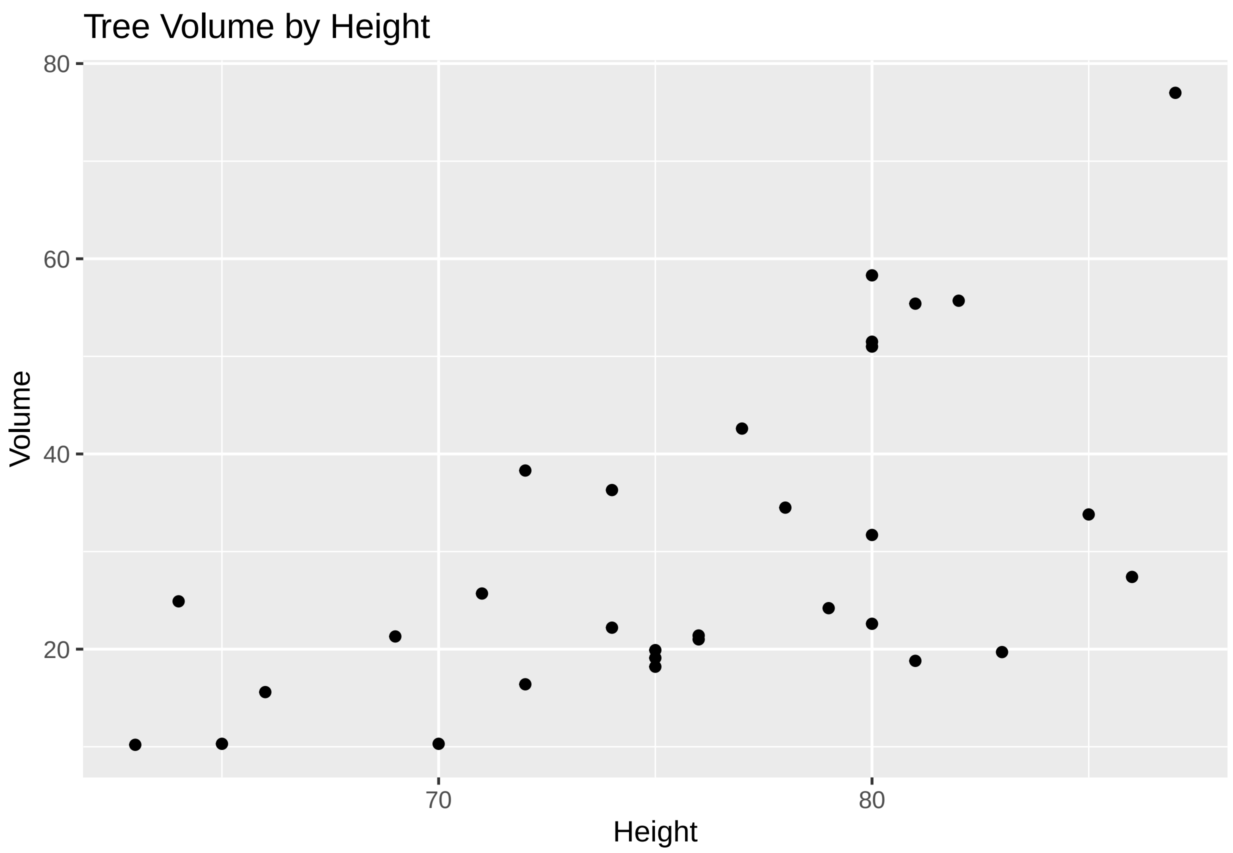
b. Using the summary command, get the y-intercept and slope of the regression line.

```
summary( tree.model )
```

```
##
## Call:
## lm(formula = tree.data$Volume ~ tree.data$Height)
##
## 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 **
## tree.data$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
```

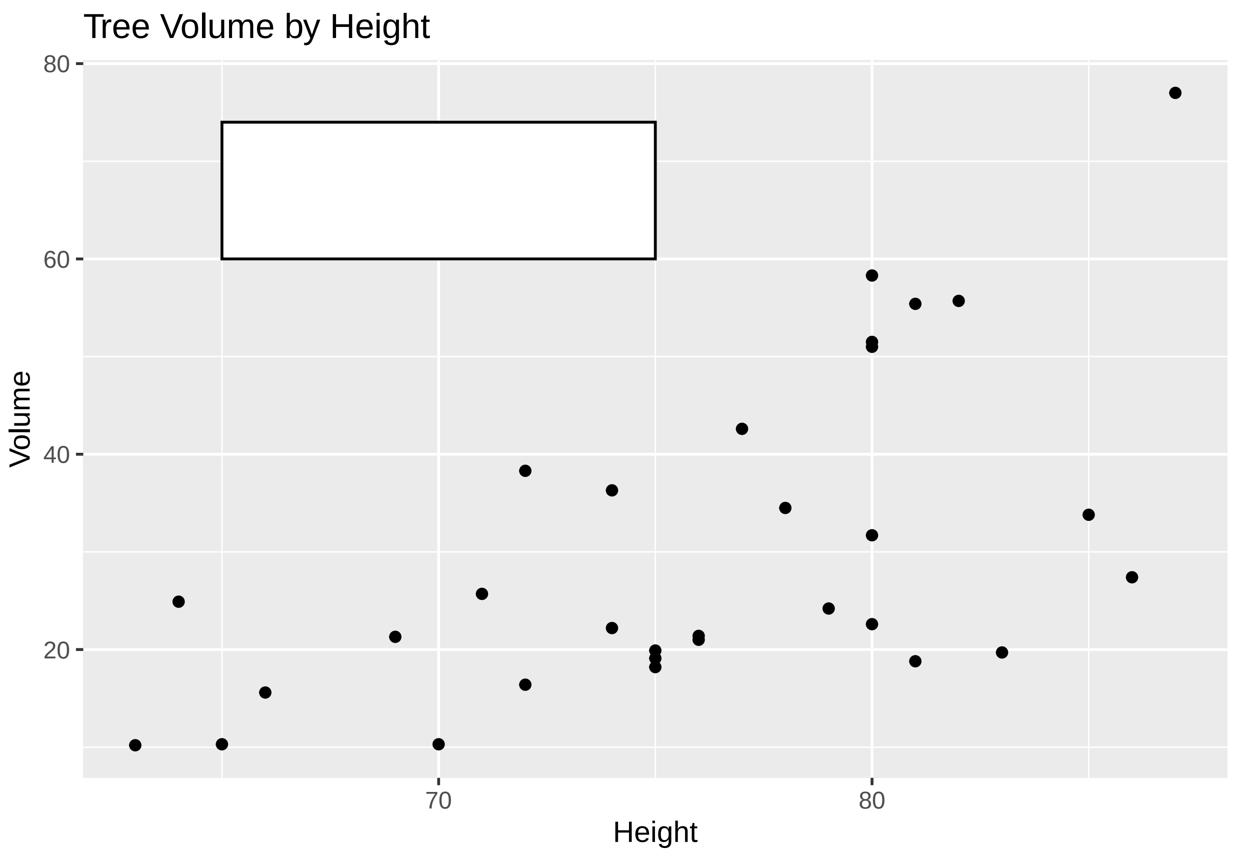
c. Using ggplot2, create a scatter plot of Volume vs Height.

```
ggplot( data=tree.data, aes( x=Height, y=Volume ) ) +
  geom_point() +
  labs( title="Tree Volume by Height" )
```



d. Create a nice white filled rectangle to add text information to using by adding the following annotation layer. annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74, fill='white', color='black') +

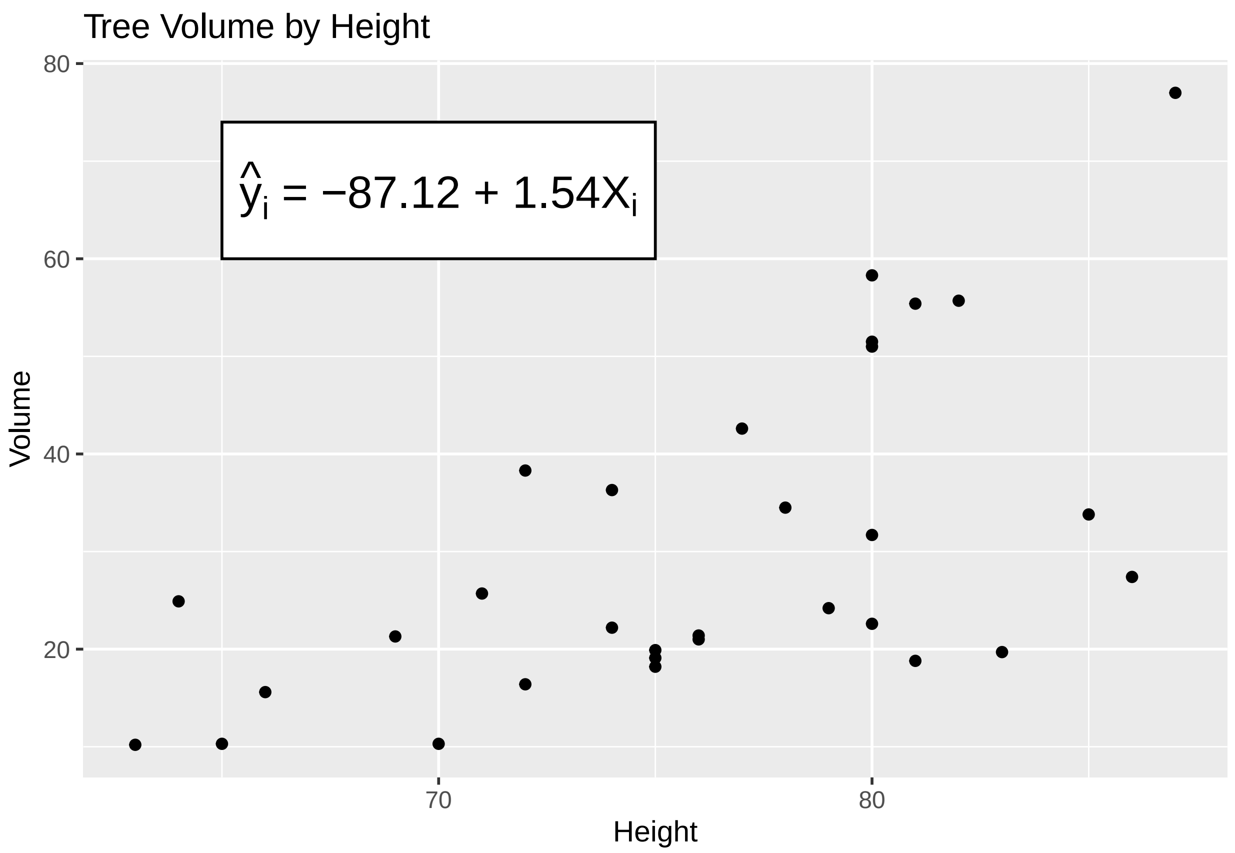
```
ggplot( data=tree.data, aes( x=Height, y=Volume ) ) +
  geom_point() +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74, fill='white', color='black') +
  labs( title="Tree Volume by Height" )
```



e. Add some annotation text to write the equation of the line yi=-87.12+1.54*xi in the text area.

```
ggplot( data=tree.data, aes( x=Height, y=Volume ) ) +
  geom_point() +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74,
           fill='white', color='black' ) +
  annotate("text", x=70, y=67,
           label = latex2exp::TeX( '$\\hat{y}_i$ = -87.12 + 1.54$X_i$' ),
           size = unit( 6, "pt" ) ) +
  labs( title="Tree Volume by Height" )
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```

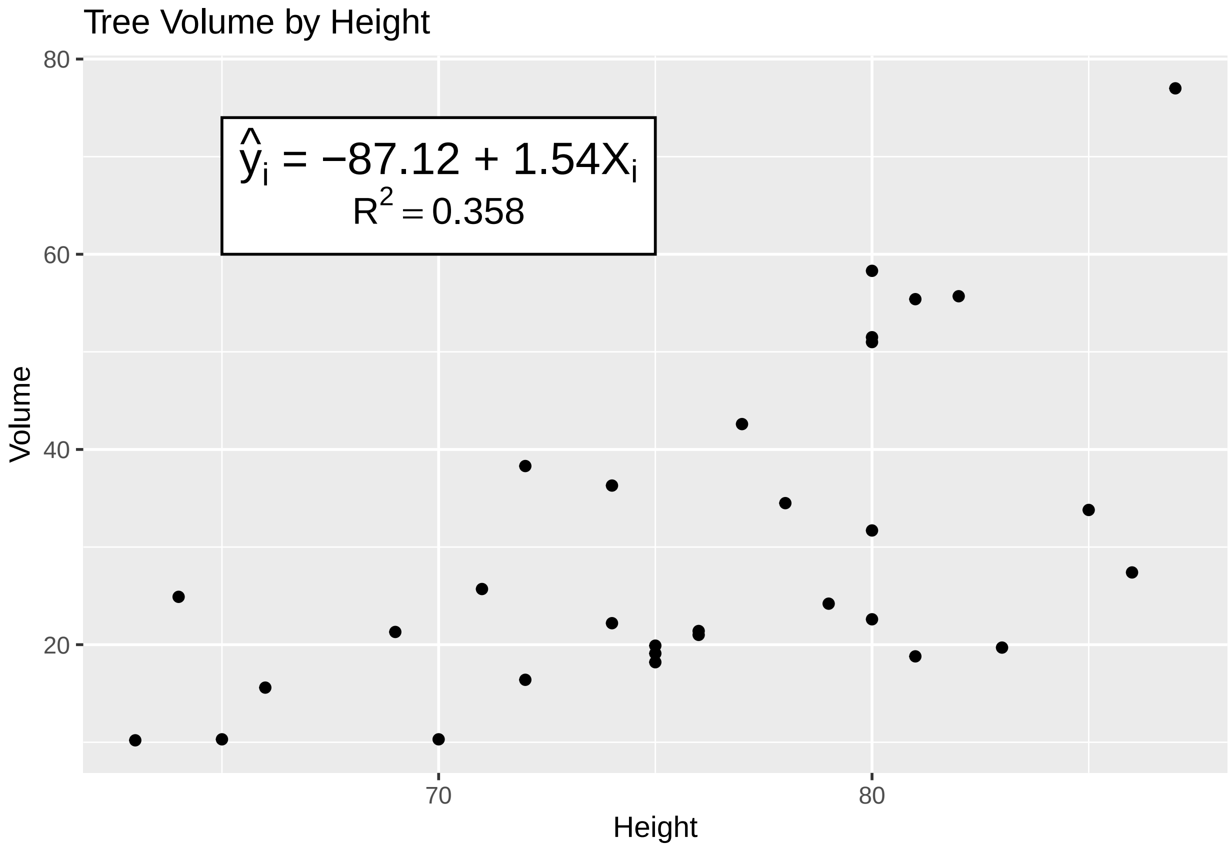


f. Add annotation to add R2=0.358

```
ggplot( data=tree.data, aes( x=Height, y=Volume ) ) +
  geom_point() +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74,
           fill='white', color='black' ) +
  annotate("text", x=70, y=70,
           label = latex2exp::TeX( '$\\hat{y}_i$ = -87.12 + 1.54$X_i$' ),
           size = unit( 6, "pt" ) ) +
  annotate("text", x=70, y=65,
           label = latex2exp::TeX( '$R^2 = 0.358' ),
           size = unit( 5, "pt" ) ) +
  labs( title="Tree Volume by Height" )
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```

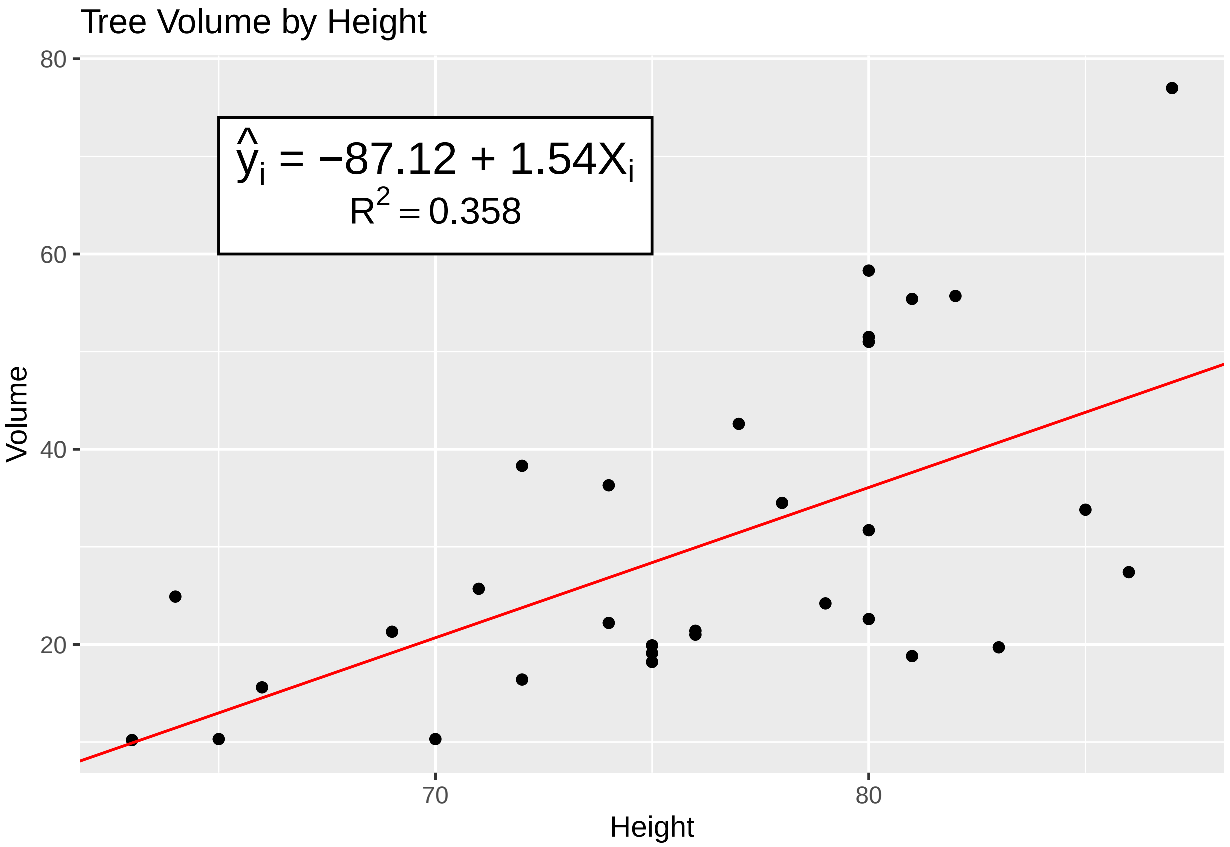


g. Add the regression line in red. The most convenient layer function to uses is geom_abline(). It appears that the annotate doesn't work with geom_abline() so you'll have to call it directly.

```
ggplot( data=tree.data, aes( x=Height, y=Volume ) ) +
  geom_point() +
  annotate('rect', xmin=65, xmax=75, ymin=60, ymax=74,
           fill='white', color='black' ) +
  annotate("text", x=70, y=70,
           label = latex2exp::TeX( '$\\hat{y}_i$ = -87.12 + 1.54$X_i$' ),
           size = unit( 6, "pt" ) ) +
  annotate("text", x=70, y=65,
           label = latex2exp::TeX( '$R^2 = 0.358' ),
           size = unit( 5, "pt" ) ) +
  geom_abline( intercept=-87.12, slope=1.54, colour="red" ) +
  labs( title="Tree Volume by Height" )
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'

## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```


Tree Volume by Height
$\hat{y}_i = -87.12 + 1.54X_i$
$R^2 = 0.358$
Volume
80
60
40
20
70
80
Height