STA_445_Assignment 5

Nicole Sylvester

2023-11-09

Exercises

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 The information is relatively out of date (from 1970s?), but will be fun to graph.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 $\log_{10}$ 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.

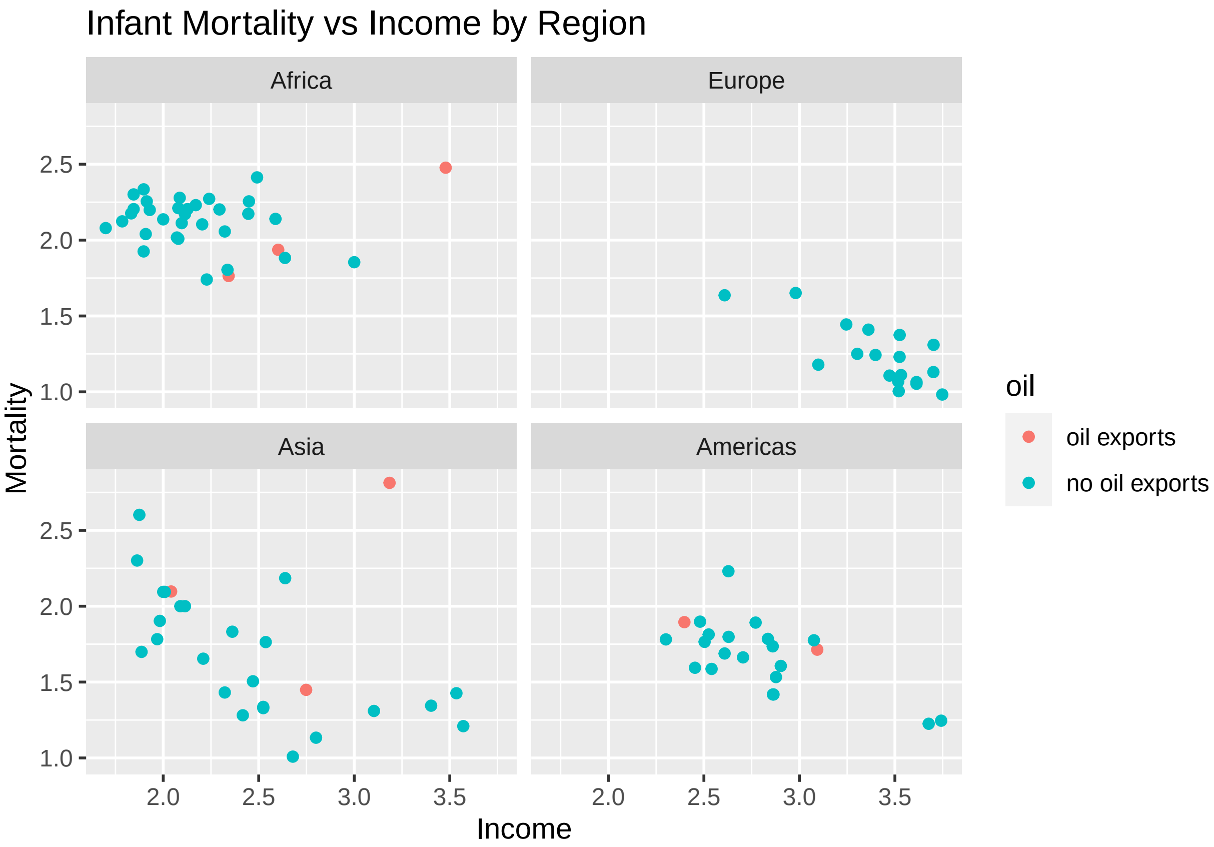
   a) The `rownames()` of the table gives the country names and you should create a new column that contains the country names. *`rownames`

```
data('infmort', package = 'faraway')

infmort$Country <- rownames(infmort)
```

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

```
P <- ggplot(infmort,
            aes(x = log10(income), y =  log10(mortality),
                color = oil)) +
  geom_point() +
  facet_wrap(~region) +
  labs(x = "Income", y = "Mortality",
       title = "Infant Mortality vs Income by Region")
P
```

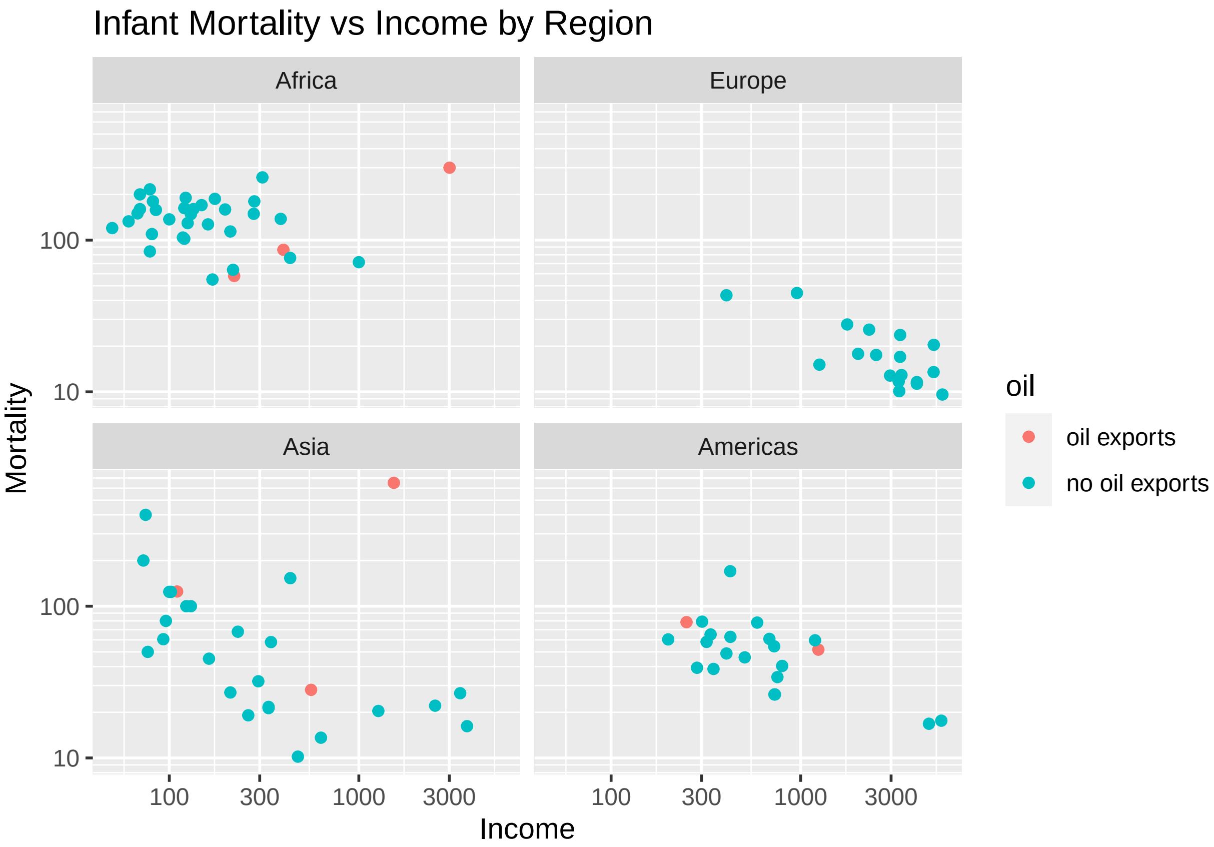
```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



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.

```
P <- ggplot(infmort,
            aes(x = income, y = mortality,
                color = oil)) +
  geom_point() +
  facet_wrap(~region) +
  scale_x_log10() +
  scale_y_log10(breaks=c(1,10,100),
                minor=c(1:10,
                        seq( 10, 100,by=10 ),
                        seq(100,1000,by=100))) +
  labs(x = "Income", y = "Mortality",
       title = "Infant Mortality vs Income by Region")
P
```

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



I

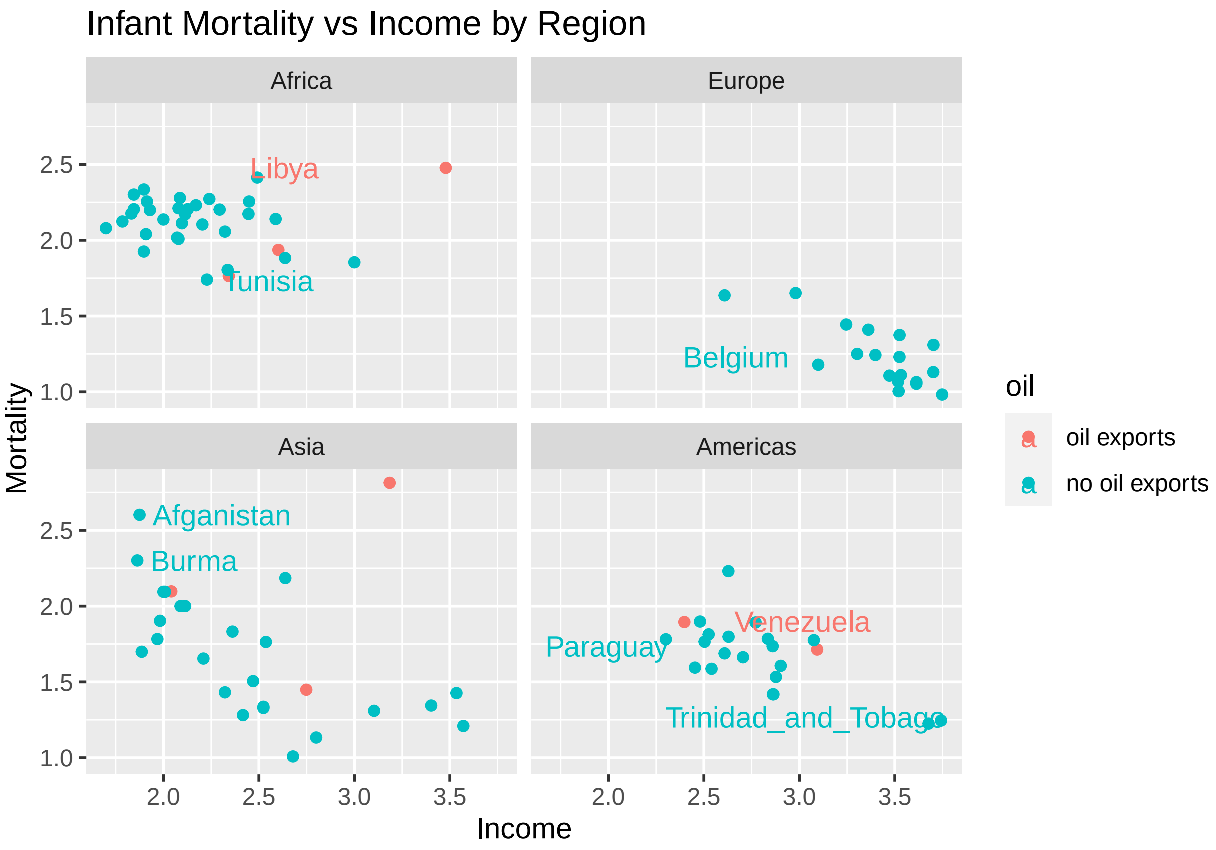
like log10 in the aes better because the smaller x and y values are easier to read.

```
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.
```

```
P <- ggplot(infmort, aes(x = log10(income), y = log10(mortality),
                color = oil)) +
  geom_point() +
  facet_wrap(~region) +
  labs(x = "Income", y = "Mortality",
       title = "Infant Mortality vs Income by Region")

countries <- sample(infmort$Country, 15)

labeled <- P +
  geom_text_repel(
    data = subset(infmort, Country %in% countries),
    aes(label = Country)
  )

labeled
```

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

```
## Warning: ggrepel: 7 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



2. Using the `datasets::trees` data, complete the following:
   a) Create a regression model for $y =$ `Volume` as a function of $x =$ `Height`.

```
data(trees)

model <- lm(Volume ~ Height, data = trees)

model
```

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

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

```
summary(model)
```

```
##
## 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
```

```
yIntercept <- "-87.1236"
Slope <- "1.54"

yIntercept
```

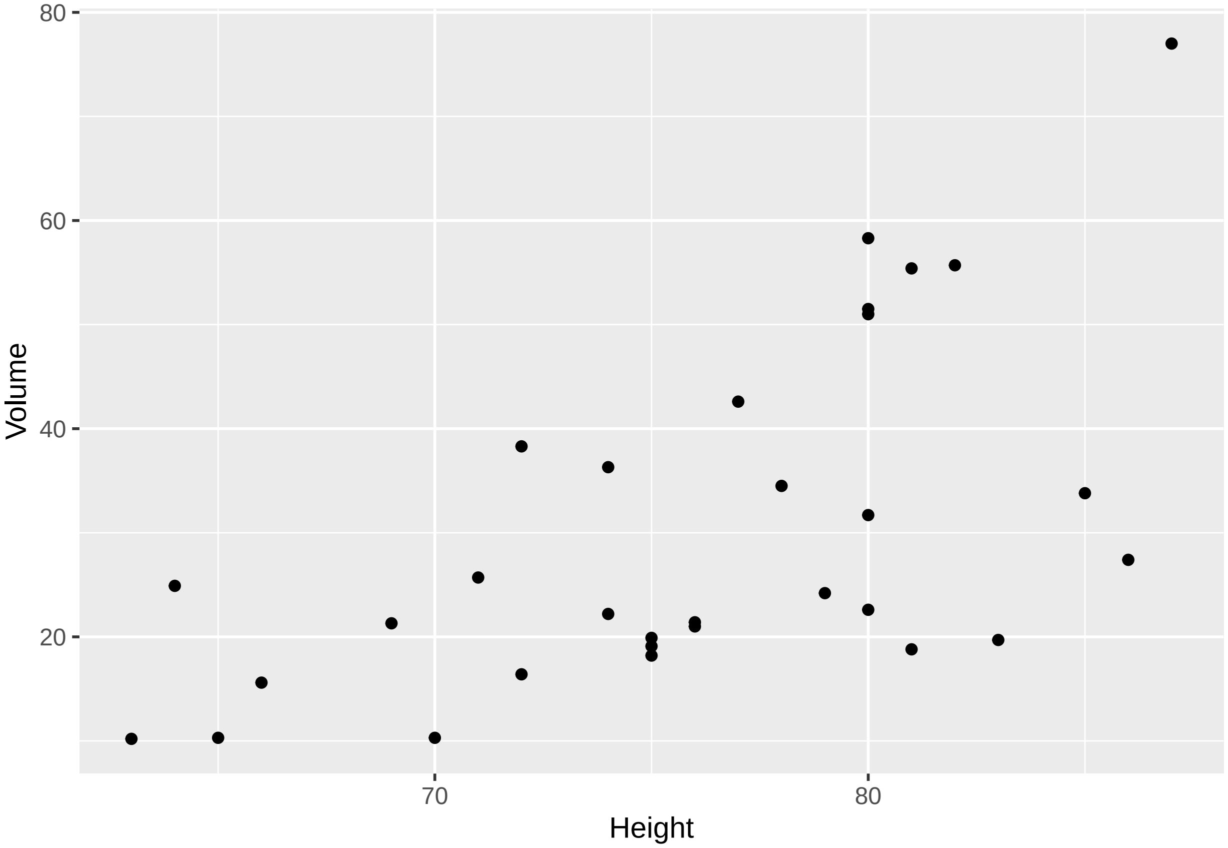
```
## [1] "-87.1236"
```

```
Slope
```

```
## [1] "1.54"
```

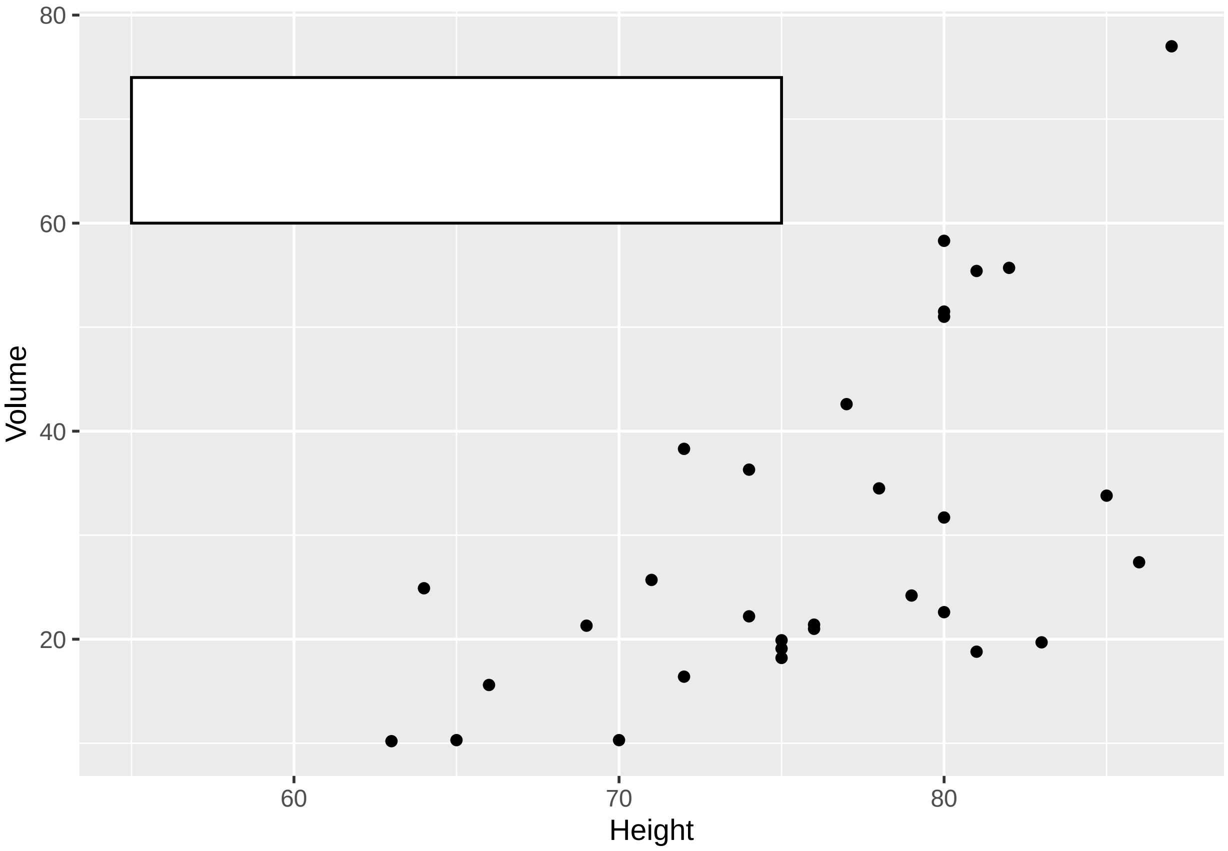
c) Using `ggplot2`, create a scatter plot of Volume vs Height.

```
ggplot(data = trees, aes(x = Height , y = Volume )) + geom_point()
```



d) Create a nice white filled rectangle to add text information to using by adding the following annotation layer.

```
ggplot(data = trees, aes(x = Height, y = Volume)) +
  geom_point() +
  annotate('rect', xmin=55, xmax=75, ymin=60, ymax=74,
               fill='white', color='black')
```

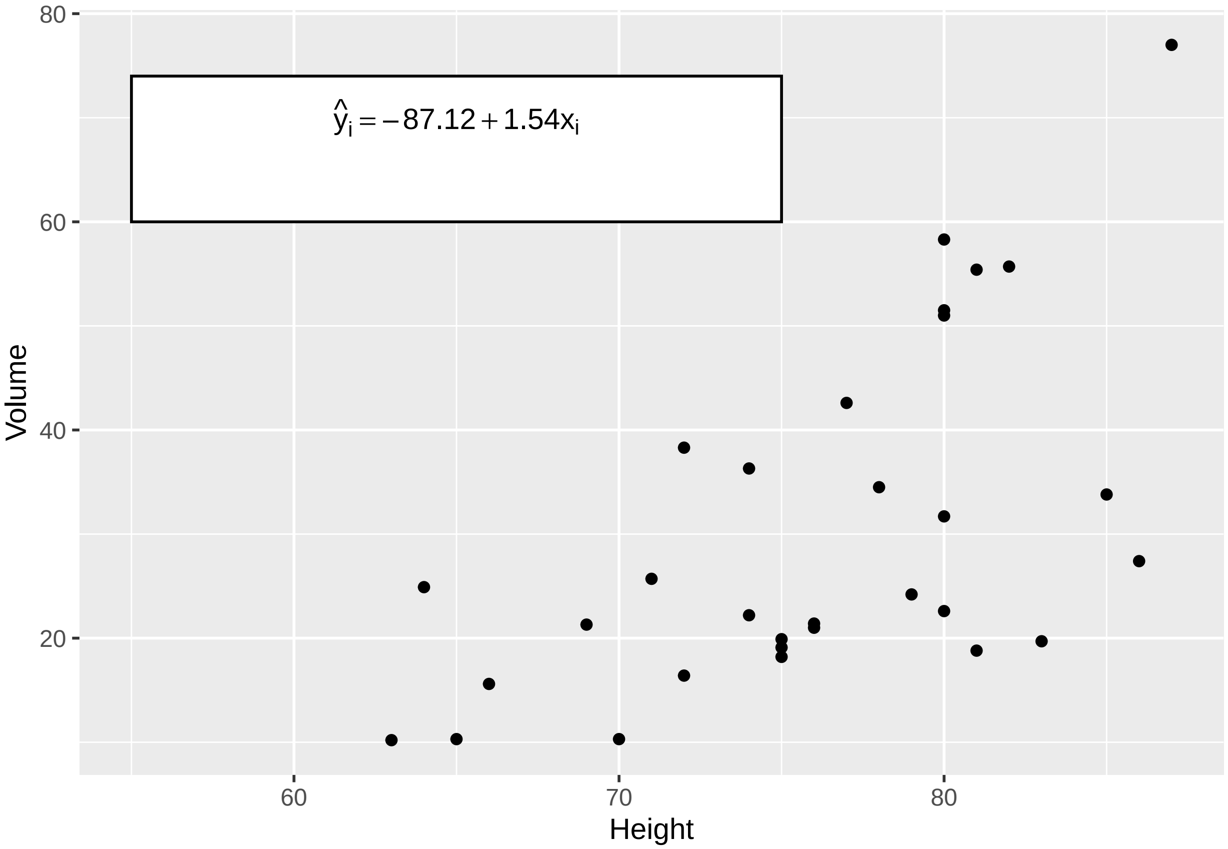


e)

Add some annotation text to write the equation of the line $\hat{y}_i = -87.12 + 1.54 * x_i$ in the text area.

```
ggplot(data = trees, aes(x = Height, y = Volume)) +
  geom_point() +
  annotate('rect', xmin=55, xmax=75, ymin=60, ymax=74,
               fill='white', color='black') +
  annotate("text", x = 65, y = 70,
         label = expression(hat(y)[i] == -87.12 + 1.54 * x[i]))
```

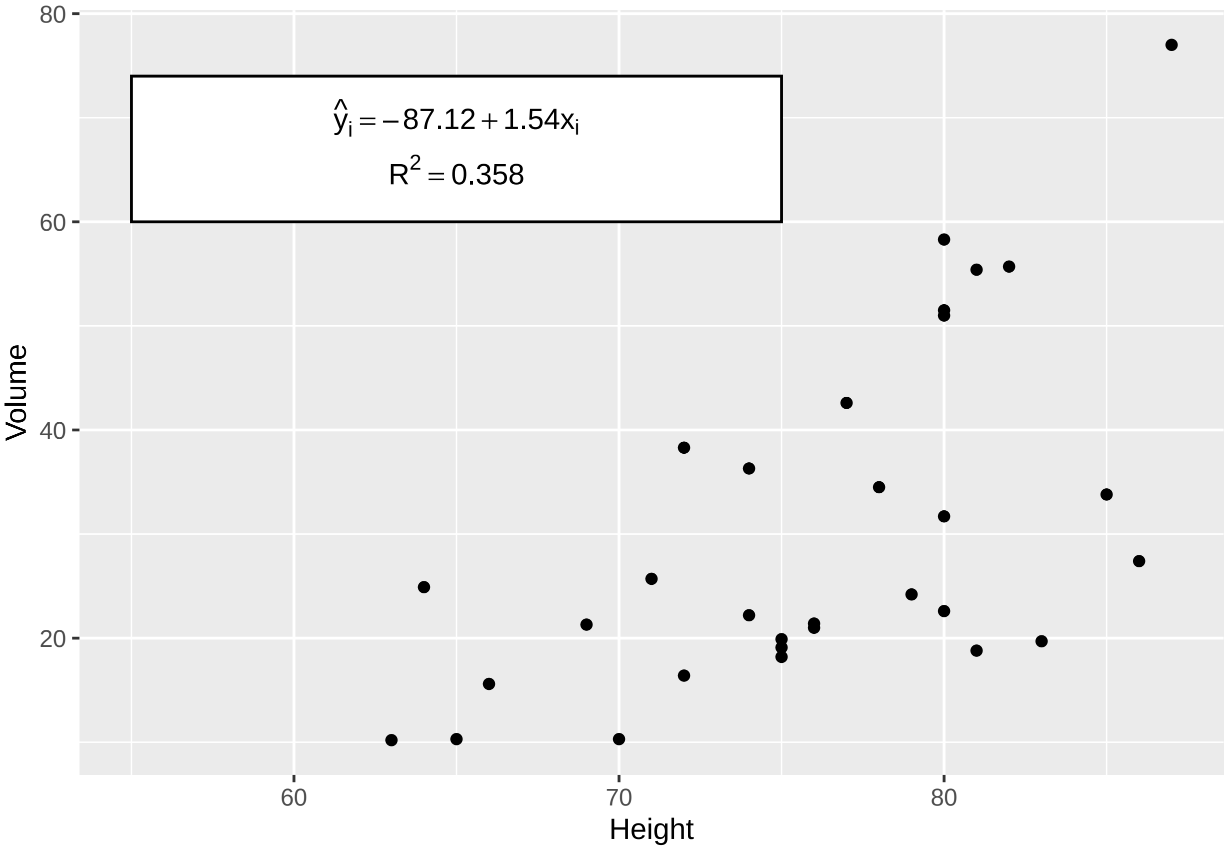
```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'
```



```
f)  Add annotation to add $R^2 = 0.358$
```

```
ggplot(data = trees, aes(x = Height, y = Volume)) +
  geom_point() +
  annotate('rect', xmin=55, xmax=75, ymin=60, ymax=74,
                fill='white', color='black') +
  annotate("text", x = 65, y = 70,
           label = expression(hat(y)[i] == -87.12 + 1.54 * x[i])) +
  annotate("text", x = 65, y = 65, label = expression(R^2 == 0.358))
```

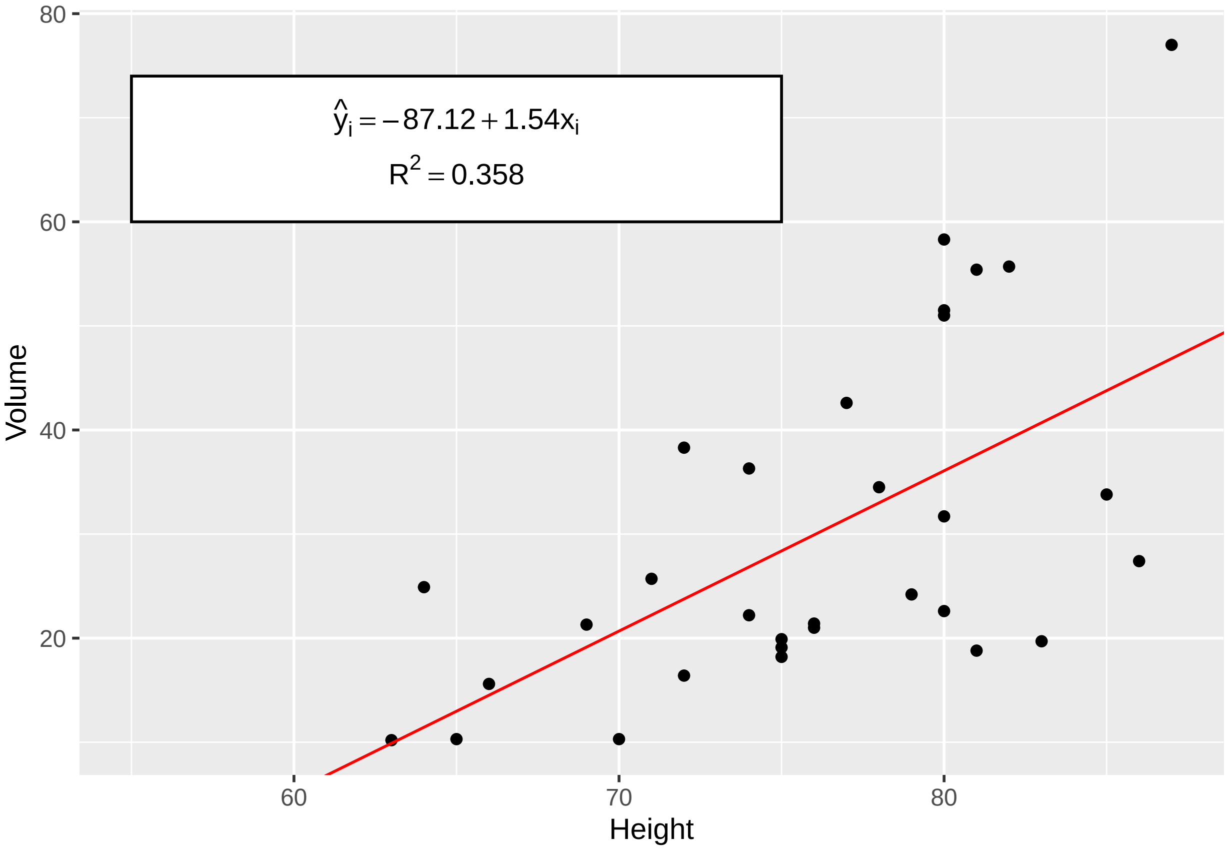
```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'expression'

## 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 = trees, aes(x = Height, y = Volume)) +
  geom_point() +
  geom_abline(intercept = -87.12, slope = 1.54 ,color = "red") +
  annotate('rect', xmin=55, xmax=75, ymin=60, ymax=74,
               fill='white', color='black') +
  annotate("text", x = 65, y = 70,
           label = expression(hat(y)[i] == -87.12 + 1.54 * x[i])) +
  annotate("text", x = 65, y = 65, label = expression(R^2 == 0.358))
```

```
## 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'
```


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